Communications of the Association for Information Systems

Volume 40 Article 12

4-2017

Applying a Layered Framework to Disaster Recovery

Corey Baham
Oklahoma State University, corey.baham@okstate.edu

Andres Calderon
Baton Rouge

Rudy Hirschheim
Louisiana State University

Communications of the Association for Information Systems

Research Paper ISSN: 1529-3181

Applying a Layered Framework to Disaster Recovery

Corey Baham
Oklahoma State University
Stillwater, Oklahoma
United States
corey.baham@okstate.edu

Andres Calderon
Baton Rouge, Louisiana
United States

Rudy Hirschheim
Louisiana State University—Information Systems and Decision Sciences
Baton Rouge, Louisiana
United States

Abstract:

Building highly available information technology (IT) infrastructures has become critical to many corporations' survival. However, the disaster recovery (DR) industry lacks a common enterprise framework to capitalize on the value that DR provides corporations due in part to inadequate conceptual frameworks for DR that can facilitate the alignment of corporate efforts toward corporate resiliency. To address this problem, we propose a new conceptualization for the DR of enterprise architecture. This conceptual framework comprises DR layers that describe the nature of DR and its related components from a functional and technical point of view. We discuss the benefits of these DR layers to DR teams and compare our approach to traditional thinking. Further, we present a case study, its findings, and their implications for DR. As a result, we demonstrate how our layered framework of enterprise architecture provides a unified understanding of the DR practice, which one can then use to support decision making and corporate alignment of the DR practice and its associated technology.

Keywords: Disaster Recovery, Disaster Response, Layered Framework.

This manuscript underwent peer review. It was received 03/10/2016 and was with the authors for 7 months for 2 revisions. Julie E. Kendall served as Associate Editor.

1 Unifying Disaster Recovery

The disruption that disasters cause represent one of the most serious information technology (IT) interruptions. In relation to data, these interruptions exist in many forms such as severe power outages, data corruption or loss, and long-term data center loss. According to a survey by PricewaterhouseCooper, 70 percent of companies that experience a major data loss actually go out of business within a year (Ireson, 2009). Although often overlooked, disaster recovery (DR), a subset of business continuity management (BCM) (Herbane 2010), has gained more attention as firms consider the consequences of disruption such as massive infrastructure damage and untimely data loss. In the past, firms have attempted to mitigate risks through careful planning, risk detection, and incident-management strategies. Unfortunately, the levee breach following Hurricane Katrina in 2005 (TAB Service Company, 2015) and many other catastrophic disasters clearly demonstrate the failures of our current response and recovery practices and highlight our inability to forecast and manage the risk associated with cascading consequences.

According to prior research, before a catastrophe occurs, professionals and the public alike can predict the problems that will arise: 1) communications systems will fail, 2) command and control structures will fracture, and 3) resources will be deployed slowly (Donahue & Tuohy, 2006). These deficiencies highlight many of the shortcomings in the current DR[1] practice and point to the lack of a unified understanding of it. Indeed, situations in which organizations have replaced their command and control structures via improvisation and bricolage (Weick, 1993) during the chaos that has ensued when a catastrophe has struck (Donahue & Tuohy, 2006) evidence this lack of understanding. In this paper, we explore a novel conceptualization of DR by focusing on the need for a common enterprise framework for understanding it. We create an effective DR framework for recovering information systems (IS) that provides a common language for the various stakeholders involved in DR so that organizations can consistently implement technologies and processes to support DR and corporate resiliency. Using a unified framework for DR, all stakeholders at organizations can: 1) improve communications through a shared common understanding of DR, 2) naturally align the application of recovery resources through patterns of well-understood operational behavior, and 3) foster repeated use of DR artifacts as an integral part of IT services and operations.

Drawing on prior research (Brooks, Bedernjak, Juran, & Merryman, 2002; Lindstrom, Samuelsson, & Hägerfors, 2010), we apply a layered approach to DR to improve how organizations recover from disasters. Furthermore, we use the term “DR orchestration” to describe the coordination of resources that an organization uses to resume its business after a disaster has occurred. In particular, we address the following research question:

> **RQ:** How does a layered approach to disaster recovery improve disaster recovery orchestration?

We argue that a layered approach to conceptualizing DR in organizations provides a useful way to understand the practice across the entire business landscape. Not only IT professionals and engineers but also business decision makers should find this conceptualization useful because it tackles the issue of understanding DR’s dependencies aligned alongside supporting technologies. Although other DR approaches may exist in practice, this approach is new in the sense that it fills a gap in DR by providing the ability to contextualize detailed plans to recovery priorities and technology dependencies. To our knowledge, no framework exists in the literature to understand DR in general.

This paper proceeds as follows: in Section 2.1, we review the extant literature. In Section 2.2, we discuss the use of a layered approach for DR thinking and present a novel conceptualization for DR in Section 3. In Section 4, we present a case study, which demonstrates the ability of a layered approach to unify the current DR practice in a large company. Finally, in Section 5, we discuss the applications and benefits of our framework.

[1] In this study, we use the term “DR” to refer to the disaster recovery of information systems and information technology and not physical structures such as doors, windows, and walls.

2 Background

2.1 BCM/BCP and DR Literature

Business continuity management (BCM) is a proactive approach to ensure that disasters do not interrupt business operations (Herbane, 2010; Herbane, Elliott, & Swartz, 2004). In this vein, Business continuity planning (BCP), a subset of BCM, refers to "an integrated set of procedures and resource information that is used to recover from a disaster that has caused a disruption to business operations" (Barnes, 2001, p. 39). In this study, we focus on the DR literature, which we define as a subset of BCM that focuses on the process of "creating and executing a plan for how an organization will resume partially or completely interrupted information technology, organizational, or business critical functions within a predetermined time after a disaster or disruption has occurred" (Lawler, Szygenda, & Thorton, 2007, p. 2). However, we note frameworks in the larger BCM (and BCP) literature base that may apply to DR. As prior studies note, the recovery phase is the least investigated and arguably the most poorly understood of the four phases of disasters (mitigation, preparedness, response, and recovery) commonly found in the DR literature (Rubin, 1991; Berke, Kartez, & Wenger, 1993). Few studies focus on IT recovery specifically and even fewer examine DR scenarios in real time. Still, DR and business continuity are top concerns for information technology (IT) executives (Kappleman, Mclean, Luftman, & Johnson, 2013) because of IT downtime's increasingly detrimental effects on firms' reputation, their ability to conduct business, and, ultimately, their survivability. In fact, according to a survey by Computer Associates Technologies, IT downtime cost firms over US$26 billion in lost revenue in 2010 (Harris, 2010). Unfortunately, methods of IS recovery largely focus on using IS recovery tools and acquiring sophisticated hardware but neglect the wider context concerning how these tools integrate with existing processes. Additionally, the DR literature focuses on modeling disaster response scenarios (Beamon & Kotleba, 2006; Bryson, Millar, Joseph, & Mobolurin, 2002; Najafi, Eshghi, & de Leeuw, 2014; Pidd, de Silva, & Eglese, 1996) or on data recovery. For example, Holdman, Kavuri, Rassbach, Hammett, and Ward (2007) developed a data DR system that provided redundancy using virtual snapshots of data libraries.

The DR literature shows that traditional approaches to DR focus on the developing procedural, task-driven plans. These plans contain a portfolio of separate tasks that one should enact immediately after a disaster occurs (Fallara, 2003). For instance, Sandhu (2002) notes several DR tasks such as determining the location of backups and identifying alternative sites. Both the DR literature and practice still lack holistic approaches for conceptualizing and managing these tasks despite their frequency in DR practice (McEntire & Fuller, 2002; McEntire, Fuller, Johnston, & Weber, 2002; Chen, Sharman, Rao, & Udaphyaya, 2013). Therefore, organizations seldom execute their task-driven plans as planned and abandon them as unpredictable challenges interrupt linear procedures.

In comparison, the BCM literature contains several frameworks, most of which focus on the development of project plans including policies and procedures that key stakeholders should act on. For instance, Gibb and Buchanan (2006) developed a framework for designing, implementing, and monitoring BCM. Their framework comprises multiple input-process-output frameworks that map to nine phases of BCM. Concerning DR, the implementation phase of the framework highlights general inputs and expected outputs such as DR plans and implementation logs, respectively. Lindstrom et al. (2010) developed and tested a framework for explaining BCP to stakeholders. Their framework uses a staircase model to adapt easily to different organizations. Bajgoric (2014) defines a systematic framework for implementing business continuity management. The framework emphasizes continuous computing technologies.

Despite several helpful insights and frameworks, the DR and BCM literature still lacks explanations about how to improve the DR efficiency of IS beyond simply fine-tuning DR plans. In addition, knowledge gains have been limited in terms of improving the efficiency of DR orchestration. In fact, prior research suggests that lessons learned have either been ignored or too isolated to generalize (Donahue & Tuohy, 2006). Additionally, according to Donahue and Tuohy (2006), the uncertainty and infrequency of disasters makes it difficult to validate even the most sophisticated DR and BCM frameworks. Nevertheless, we need a DR approach that focuses on DR's efficiency and not just BCM's planning phase to improve DR's efficiency in practice. As such, we draw on prior research to articulate a novel approach to improving DR orchestration.

2.2 The Need for a New Understanding

Rooted in traditional military command and control (C2) doctrines (von Steuben, 1779), the DR practice assumes that decision making and authority need to be centralized despite the evolution of more recent

C2 thinking. In a military organization, C2 refers to a properly designated commanding officer's exercising authority and direction over assigned and attached forces to accomplish a mission (Department of the Army, 2011). Recent C2 studies show the impracticality of assuming that a single individual can acquire all of the necessary information to succeed in a dynamic work situation (Sonnewald & Pierce, 2000). Nevertheless, organizations continued to create task-driven plans, which have neither yielded great success nor unified DR practice (Kendall, Kendall, & Lee, 2005). Kendall et al. (2005) point out that many BCM and DR specialists resort to improvisational tactics because one cannot feasibly use large, heavy documents during a disaster. While task-level support tools can be useful, research suggests that they are only as useful as the communication and support system that they connect to (Hale, 1997). The DR practice lacks and needs a new unified support framework. As prior research evidences, using layered approaches has helped with DR thinking. For example, Hale (1997) used the open system interconnection (OSI) model as a base by which to build a layered communication architecture to support crisis response and move from descriptive tasks to a prescriptive model of crisis response support. Other studies contain layered or tiered approaches for business continuity and disaster recovery. For example, SHARE's "Seven Tiers of Disaster Recovery" highlights varying levels of recovery in mission-critical information systems (Brooks et al., 2002). The model promotes a unified understanding of the methods of recovering mission-critical computer systems by conceptualizing multiple tiers, which define current service levels and associated risks. The contributions of layered models and the fragmented view of the DR practice from an enterprise architecture (EA) perspective motivate our application of a layered framework to conceptualize EA for the DR practice.

We understand that some might view using a layered approach as simply optimizing old DR thinking instead of optimizing new DR approaches such as N+1 datacenters and active-active technology-based advancements. In active-active architectures, companion technologies are configured to run alongside primary technologies to take over a failed primary technology's devices and connections. Such criticism has some credence; however, not all corporations can maintain active-active with excess redundancy capacity for failovers. Furthermore, the vast variety of business contexts that exist, even among large corporations, show that corporations differ in their technical infrastructures and the degree of integration therein. For instance, market leaders such as Google, who own their entire technology stack (Barroso, Dean, & Holzle, 2003; Nielsen, 2015) and are more bleeding edge in how they use technology, have the means to run on three or more data centers and run uniform and controlled distributed applications that have resiliency as a core design element. As such, they can provide continuous service through automatic or near-seamless recovery, which eliminates the need for most traditional DR practices. However, the majority of large businesses do not have the means to run under a uniform, controlled, and bleeding-edge infrastructure. Furthermore, many companies consume commercial off-the-shelf (COTS) software and have to integrate systems to operate; therefore, they are tied to the traditional DR practices. Thus, organizations require a more efficient and collaborative approach to DR.

Accordingly, we propose that a layered approach built on EA provides a focal point for a unified DR framework. EA provides a platform to address multiple stakeholders using a common language as and the ability to drill down to a deeper level of detail in each division. Thus, EA provides the appropriate basis on which to juxtapose a layered approach for providing a common language for the various stakeholders involved in DR. In addition, a layered approach provides a simple framework that stakeholders can map their task-level support tools against. Moreover, prior research suggests that partitioning reduces model complexity and modulates future design (Hale, 1997).

Based on prior layered approaches for enterprise architecture (EA) (Brooks et al., 2002; Theoharidou, Kotzanikolaou, & Gritzalis, 2007; Vescoukis, Doulamis, & Karagiorgou, 2012), we conceptualize a new framework using a layered approach for DR for several reasons. First, layered approaches provide an important organizational feature: modularity. Modularity allows one to subdivide work and create a common understanding of the layers' boundaries, which provides specific elements for integrating detailed protocols. Our vendor-neutral framework characterizes and standardizes the services of the infrastructure in IT, which is represented by partitioning the technology into abstraction layers. Thereby, our framework uses a flexible, layered approach to DR, which avoids the difficulty of fitting into strict protocols.

3 A New DR Conceptualization

We create a new framework for disaster recovery that we term the "DR layers framework" (see Table 1). The second author, after experiencing multiple disasters, observed the lack of a unified understanding of DR across the industry. Together, we found the layers concept to be a helpful framework in other IT

studies for conceptualizing and mapping related concepts together. When applied to EA, it provides a common backdrop to map related technologies together in a way that is easy to understand. The framework emphasizes breaking down technology from a functional disaster-recovery perspective and not from a technology-architecture perspective. Using a layered approach that includes seven layers (from layer 1 to layer 7), we focus on understanding the priority of recovering EA and the inherent dependencies of the higher layers on the lower layers. Again, we create this framework to help organizations coordinate people, processes, and technology, and it spans both technology and the organization.

Table 1. DR Layers Framework

DR layers
7. Application
6. Virtualization
5. Physical
4. Controllers and monitoring
3. Storage and backup
2. Network
1. Data center

Our DR layers framework starts from DR layer 1 and culminates at DR layer 7. Beginning with DR layer 1, as we move up, the services offered at each subsequent layer tend to provide more business value to the organization and also experience a higher rate of change. For instance, power supplies for the core data center infrastructure in DR layer 1, which provide electrical power to the enterprise, typically remain unchanged as the organization changes the network technologies that depend on them to meet the needs of the business. At the top layer, DR layer 7, users interface with applications, which provide continuous business value and changes. In contrast, the lower layers tend to have more dependents and require more availability. We design the DR layers to maintain distinctiveness and modularity. As such, our framework has similarities to the OSI model, which addresses network interconnectivity.

Each DR layer is a functional composition of an ordered set of services, each with several sublayers. To further define the details of the proposed DR layers, we present a set of DR sublayers in Table 2. These DR sublayers may vary depending on the type of technologies a corporation uses, but this representation is a generation representation of the technologies that most large corporations currently use. For example, one could present an information system as a set of sublayer elements that are aligned and prioritized in preparing for a recovery. In a given information system, its storage might reside in sublayer 3.1, whereas its server might reside in sublayer 5.3 (physical compute) or sublayer 6.4 (virtual). The alignment and priority of these sublayer elements is essential to establishing a fluid cadence (or working rhythm) during a recovery.

These layers imply a level of dependency in that one needs to recover a layer before one can bring the next layer online. The sublayers further define the cadence of recovering technology elements. There exists a dependency and inheritance across layers, so each layer serves the layer above and depends on the layer below it. For example, the layer 2 provides the communication path needed by higher-level layers such as layer 3 (storage and backup) or layer 7 (applications). While some of these dependencies are not essential for the subsequent layers to function, organizations should carefully document exceptions across the layers and communicate them internally because these exceptions present risk downstream. For example, one could argue that rugged network switches can be brought up without HVAC in the datacenter; however, one should only do so to avoid damaging devices due to excess temperature.

While the DR layers imply that one should recover each layer before the next layer in the hierarchy, the framework does not preclude decision makers from making exceptions to the cadence priorities. There are many situations in which organizations have not recovered a lower layer's devices before a higher layer's devices due to a need to continue with the recovery process. However, these exceptions should work as situational awareness triggers that let all involved know of the exception's implications.

Table 2. DR Sublayers

DR layer	DR sublayer
Layer 7: application layer	7.6 Non-critical—tier 6 7.5 necessary—tier 5 7.4 Mission critical—tier 4 7.3 Mission critical—tier 3 7.2 Mission critical—tier 2 7.1 Mission critical—tier 1
Layer 6: virtualization layer	6.6 Virtual Desktops 6.5 Virtual desktop infrastructure 6.4 Virtual application infrastructure 6.3 Virtual production guests 6.2 Virtual deployment mgt. center 6.1 Hypervisor
Layer 5: physical layer	5.3 Physical compute 5.2 Physical file servers 5.1 Physical database servers
Layer 4: controllers and monitoring	4.5 All other information security 4.4 System monitoring 4.3 Establish Internet services 4.2 Physical directory services (i.e. AD) 4.1 Physical domain name systems (DNS)
Layer 3: storage and backup	3.4 Replication 3.3 Backup storage 3.2 Physical storage 3.1 SAN storage
Layer 2: network layer	2.4 SAN fabric 2.3 Core data network 2.2 Voice network 2.1 Network security (perimeter)
Layer 1: data center layer	1.4 Environmental (i.e., HVAC) 1.3 Power supply 1.2 Fire suppression 1.1 Safety and physical security

Often, organizations make these exceptions to optimize how they use resources or as workarounds to satisfy corporate needs. Nevertheless, the DR layers and sublayers and the implied constraints these layers establish in terms of priority and sequence of cadence should not conflict with sound judgment from technology and business management. In fact, some exceptions around the layers can present serious risk to an organization. For example, if one bypasses sublayer 2.1 (network security), there could be cyber security consequences such as data loss due to a security breach. The same applies when one decides to bypass sublayer 1.1 (safety and physical security) for sublayer 1.3 (power supply). For instance, there could be water in the datacenter when power is reestablished and, thus, a risk of electrocution. The risks and consequences of breaking the recommended cadence should serve as a signal for management to carefully analyze, coordinate, communicate, and document decisions in order to prevent undue exposure to risks.

At the top layer, the DR layer framework contains a tier-level approach to recovering applications based on the time to recover an application (recovery time objectives). In DR layer 7, each tier represents the applications that need to be available to an organization by a certain timeframe. As an example, tier 2 groups all applications that need to be available between 24 and 48 hours. The OSI model classifies the applications layer based on communication patterns, resource availability, and synchronizing communication (IOS/IEO Commission, 1994; Jain & Agrawala, 1993). In contrast, the DR layers framework generally classifies applications based on their business priority as determined by: 1) the time to recover the application and 2) the allowable data loss for that application. Since many companies rank applications based on their criticality to the business, the tier-level approach to assists companies in classifying applications as needed.

Note that one can add additional layers or sublayers in accordance with inter-layer dependencies, but doing so could complicate this simple proposed framework. We focus here not on canonizing the number of or titles of the layers but on introducing a different approach to thinking about DR. In fact, we propose a layered approach to DR over alternative methods such as implementing a connected graph of all dependencies across technologies, which tends to be complex and difficult to conceptualize in large organizations. Instead, our layered approach allows for as many sublayers as necessary to accommodate an organization's needs while keeping the overall operating picture simple. We understand that, with a layered approach, disagreement between IT practitioners concerning the number of layers is a fait accompli. For instance, the critics of the OSI model thought there should be more layers (Stein, 2004) such as a governance layer as layer 8. Why not? The point here is to recognize the gap in DR; that is the lack of a common framework for decision making, coordination, and collaboration. Thus, this research is not prescriptive, but a case study of one approach. The same should be said for additional optimization within each layer such as the prioritized testing of components based on the priority of the service they relate to. This simple framework provides the backdrop for more intense discussion of these contextual issues that may differ from firm to firm. In a DR scenario, many decisions have to be made on the fly, so a simple framework to align the decision making around business needs and technology components is imperative. Again, the layers were not developed only to establish the sequence of a DR, but as a unifying framework for collaboration and coordination across the enterprise.

While the scenario presented in this paper is not the classical earthquake, hurricane, or flood initiated disaster, the impact of the unexpected loss of power to an entire production datacenter during business hours has the same impact on the operation, regardless of root cause. In our case study, 1) teams were unable to communicate with remote sites as there was limited cellular phone connectivity inside the datacenter, 2) all systems collapsed and were at unknown states, 3) all production work at the site was interrupted, which affected teams globally, and 4) chaos immediately ensued as the recovery efforts commenced. The DR layers immediately started to guide the recovery efforts of distributed teams using a taskboard application that was made available to all responding teams for situational awareness. Teams worked around the clock to fully recover a highly interconnected and interdependent datacenter with a classic technology footprint that ranged from mainframe, middle tier, and WINTEL technologies. Our case describes how the DR Layers were used to orchestrate a full recovery of a crashed datacenter in less than 7 hours. We contend that just as Alpha was able to recovery using the DR layers, their application should help other organizations facing similar DR challenges.

4 Disaster-recovery Event

We conducted a case study to test the proposed DR layers framework in practice during a complex DR scenario at a large enterprise, which we refer to henceforth as Alpha.

4.1 Company Information

At the time of writing, Alpha was a large healthcare services organization in the United States with over 3,000 employees. In terms of IT infrastructure, Alpha had three datacenters with a wide array of technologies ranging from mainframes and midrange services to WINTEL platforms to support its offerings. Critical infrastructures at the datacenters had N+1 resilience that ensured system availability in the event of component failure. Alpha's IT team had over 300 staff that managed over 2,000 servers; maintained, designed, developed, and supported over 300 applications; and delivered IT services to Alpha. Overtime, Alpha developed a mature DR practice with comprehensive DR plans that it tested once a year and that followed a formal internal and external review process. During the time of the DR event in question, the company was transitioning from a tape-based backup recovery to an all integrated active-active and active-passive recovery strategy. This massive change was one of the primary forces behind the company's paradigmatic change and the resulting approach that we present in this paper.

Before the DR event, the company based its DR practices on command and control structures and had the classic BCM and DR processes in place including ITIL and other industry "best practices" that allowed the company to respond only limitedly to a catastrophic event. We employed a case study using participant observation in this study (Jorgensen, 1989). By observing participants, we could examine factors that would otherwise have remained concealed using other methods (McAvoy & Butler, 2007). Using this approach, we examine how we applied a layered approach to disaster recovery thinking to improve disaster recovery orchestration. The first author, whom Alpha previously employed, was familiar with the company's DR practice and observed the development and potential contribution of a layered

framework to the company's DR practice. Additionally, the second author was instrumental in developing the layered framework and participated in departmental meetings with IT managers, formal and informal discussions, and so on but also worked with (both technical and non-technical) people at all levels of the organization. After being given access to observe Alpha's DR practice, we began to collect data as we detail in Section 4.2.

4.2 Data Collection

The first author observed Alpha's DR practices for 10-20 hours per week for 12 weeks before the research team applied the DR layers framework. The observation included taking notes of incidents and practices and observing the direct and indirect influencers of Alpha's DR program. The researcher who developed the DR layers framework worked 40-50 hours per week at Alpha for approximately six months and had DR work experience at other companies. In order to produce a rich understanding of the project context, we used multiple additional data sources to triangulate the findings including semi-structured and face/face interviews, DR documentation analysis, DR department meetings, and a post mortem (see Table 3) at Alpha (Dubé & Paré, 2003; Eisenhardt, 1989; Yin, 2003). In total, we conducted formal interviews with one DR manager, two DR contractors, and 1 agile coach that ranged from 30-60 minutes. We gathered reflective feedback immediately after applying the DR framework during the DR event via a project post mortem questionnaire. We gathered the feedback from IT executives that participated in the DR events, which we used to validate the DR framework. All participants were IT executives with over 10 years of experience managing IT projects. The work we present here is part of a broader three-part study on improving the efficiency of DR orchestration. This particular study focuses on the layered framework that we developed and later applied to organize the backlog of a kanban-inspired task board that Alpha used during the DR event. Kanban is a "pull" approach that enables members to select work items from a project backlog with few restrictions (Wester, n.d.). The nuances of the kanban methodology and its applicability to DR orchestration are the focus of another study. However, we highlight the application of the DR layers framework using the taskboard application in this paper.

Table 3. Data Collection Overview

Method	Source
Direct observation	DR manager, DR contractors
DR Documentation analysis	DR paperwork
Group meeting: DR department	DR manager, DR contractors (1 hour each) Agile coach, DR manager (30 min.)
Interview	Agile coach (1 hour)
Post mortem: first event	DR manager Director, IT network operations center (NOC) Systems engineering architect SUPV, IT-computer and NOC operations NOC engineer Sr. NOC engineer Manager, IT production supply and NOC Manager, IT-systems support

4.3 Data Analysis

We transformed all data sources into textual form. We audio recorded all interviews and immediately transcribed them. We identified various themes as the study progressed as we confirmed or disconfirmed our initial observations. Although we found other themes, the effectiveness of the layered approach over past approaches to DR at Alpha remained the center focus. Thus, the first author used open coding to categorize the data into separate units of meaning, each with its own code. We compared and contrasted the units with each other and consolidated them into broader units until we reached saturation (Orlikowski 1993).

4.3.1 The DR Scenario

On August 9, 2014, a datacenter at Alpha experienced an unexpected power loss due to the failure of automated transfer switched systems at the hosting provider site. Alpha activated its DR plan and notified teams of the event. At Alpha, the DR department worked in conjunction with more than 15 subunits of IT to create recovery sequences, called a DR cadence, which reduced the recovery plan into smaller actionable components that one could assign to a team or an individual to execute. The DR cadence contained fields such as activity ID, layer ID, activity type (shutdown, recovery, validation, communication, etc.), activity description (specific system or environment), team responsible, primary subject matter expert, and so on. Alpha's IT subunits used the DR cadence as a starting block for the recovery. By assigning a layer ID to each activity, teams could map their tasks to the corresponding layer and sublayer of the DR framework that the activity affected. Thus, Alpha used the DR layers framework to organize these actionable components from the DR cadence across the enterprise.

Next, we made the DR layers framework available to team members using a simple, taskboard application. This application allowed the DR manager to upload tasks from the DR cadence and visualize the flow of work while maintaining situational awareness across the IT department. The DR cadence made up the initial recovery backlog, which is listed under the "To Do" column (see Figure 1), while the DR manager used the DR layers framework to organize the recovery backlog and signal the DR layer that would be the focus of the next batch of work items using the "up next" column. The DR manager inserted the IT groups that played a vital role at the specific layer into the inner lanes, which columns 1-4 list. At Alpha, multiple teams managed the technologies in a particular layer, so the DR manager labeled the lanes according to those teams. For instance, the network team managed most of the network technologies, so they took the initiative of pulling work items from the backlog at the stage of the recovery that required establishing network connectivity. However, input from the server team was needed during network layer activities to verify that servers were replicating correctly. Additionally, the other teams could visualize and interact with the network team as needed (see Figure 1).

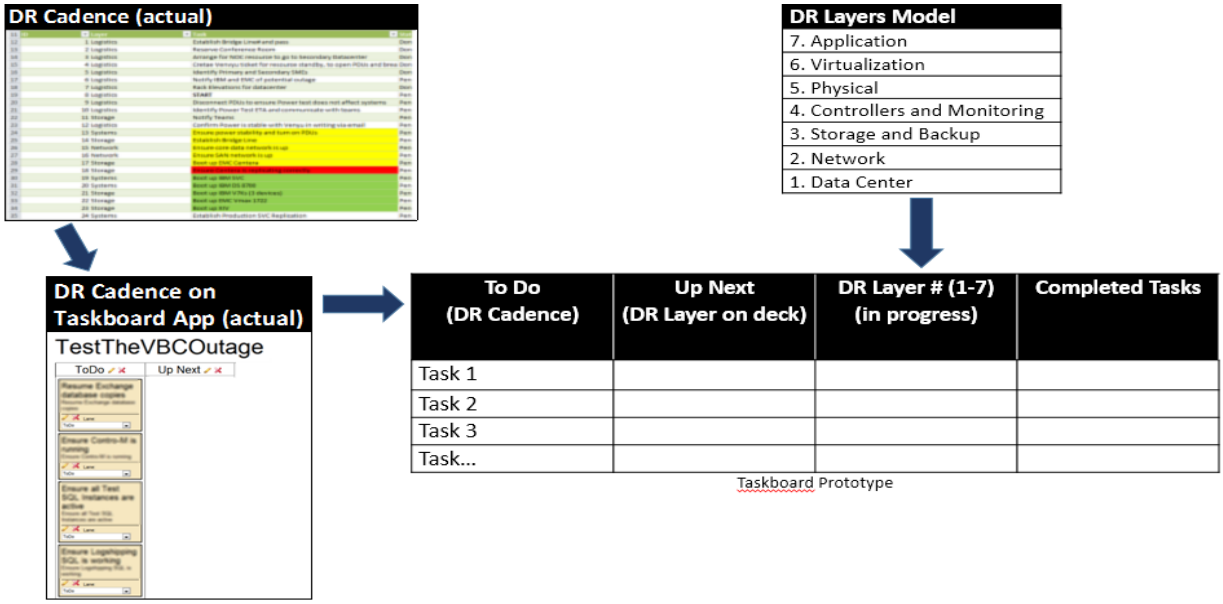

Figure 1. DR Layers Implementation

The DR team determined its objectives according to the actual remaining backlog to recovery, the dependencies of these activities, resources at hand, input from leadership in setting priorities, risk mitigation, and other realities that affected the recovery. Using the DR layers framework, pertinent stakeholders from across the organization could gain a common understanding, which the detailed plans did not provide, especially during a disaster. As a result, the DR teams successfully recovered a total datacenter collapse in less than seven hours, which included validating applications. According to the DR manager, prior to completing the aforementioned recovery, the company could not update its documents

[2] We blur Alpha's company data for confidentiality.

to keep pace with its changes in personnel, processes, and technology, which placed the company in jeopardy of attempting an IS recovery using outdated information contained in the runbooks. Such was the company's impetus for seeking an alternative framework that could overcome the shortcomings of traditional DR approaches. In Section 4.3.2, we highlight the results of using the DR layers framework.

4.3.2 Results

Using our layered approach offered many enhancements over using a traditional task-level approach alone. First, the layered approach encouraged Alpha to map DR activities against a common backdrop, which promoted a common understanding. In traditional document-driven approaches, a DR plan document, though comprehensive, is monolithic, not up to date, and inflexible to adapt, and it generally assumes that the core teams are available and can communicate the specific response across the enterprise. Previously Alpha used recovery sequences or "runbooks" that detailed every step of the recovery process for each critical application, but did not have detailed documentation on how to recover underlying technologies at the lower layers of infrastructure since the infrastructure documents resided with those outside of the assumed core teams. Alpha reviewed its DR plan at least once a year as part of the yearly DR test and used it only for disaster-recovery purposes (as disability insurance to a disaster). Further, the DR plan did not ingrain vital operational changes and, consequently, tended to go stale within weeks. Most importantly, the recovery plans for application used by each functional area lacked details concerning the dependencies of these application on systems in other functional areas. For example, the DR plan did not have all the details about the network configuration: it had only the "necessary" high-level network information to support a limited set of recovery scenarios. Using the layered approach allowed the organization to have each layer with its full reference documentation. Alpha used and maintained this document, which included all pertinent details necessary to manage and restore that layer, as part of the operation. The comprehensive, familiar, and relevant documentation was there for recovery if needed, which was extremely valuable since disasters generally do not follow planning scenarios and, as complexity increases due to unforeseen consequences, the related documentation was available to support the responding teams.

Conversely, our layered approach allowed Alpha to better understand the dependencies that existed across systems while making investments in DR. It could do so because the layered approach provides a way to divide and visualize densely connected resources in the EA into functional "modules" and allows stakeholders and teams to focus DR efforts towards sets of highly related tasks and priorities. On this point, the DR manager commented:

> *The DR layers framework helps communicate to the business the dependency nature of the technology. By breaking the DR plan into logical elements, it is a lot easier to have an orchestration. It is a lot easier to (handle) the dynamic nature of decision making. It is a lot easier to adjust to last minute changes.*

Furthermore, staff members at all levels of the company and across all functional areas used the understanding of the DR practice and associated dependencies to communicate with each other. The proposed DR layers provided a common language by which staff members could communicate their recovery activities and contribute to the recovery with minimal friction. The DR manager could easily communicate with all functional divisions of Alpha and provide situational awareness using the layers with the taskboard application to provide a tacit but clear status message to all. The network operations center supervisor stated:

> *The board helps with a visual of the multiple tasks assigned and interdependencies. Before the board…, the project manager [would] contact and gather the appropriate team members together should an issue come up which is a longer process and takes longer to resolve issues with a traditional cadence project.*

Second, the DR layers approach provided a high level of visibility and transparency to the DR efforts, which helped align teams and corporate efforts and allowed everyone to plan around the layers. In the past, the DR manager was bombarded with inquiries as he was attempting to adapt static plans and maintain all informed on next steps. Calls (mostly voicemail since phones were down in the power loss event), emails, and text messages poured in from executives who requested frequent updates concerning the effect of the disaster on their department and projects. As a result, Alpha had to abandon the DR plan in its original form as the DR manager started organizing the recovery, identifying which personnel or vendors could assist in restoring the systems, and gaining a better understanding of the event's impact on

the organization's people, technology, data, and operations. In general, detailed plans are replaced by improvization and bricolage as chaos ensues; once the dust settles, one must adapt plans to meet whatever the circumstances require. As such, the distributed efforts of teams need to unify around a common understanding of how to restore order that is ingrained in their DR work culture and guided by a well understood DR framework.

While the DR layers approach provided the DR manager with a structure in which he communicated from in amid the chaos, the distributed recovery efforts could independently rally the recovery. For example, one can estimate how long it will take to recover each layer. Since the automated taskboard supports the layered approach, the approach clearly keeps an activity record of the recovery efforts, which allows teams to accurately forecast how long a resolution will take and allows them to better predict when to expect applications to be ready for use. As such, BCM personnel and decision makers can better plan for their contingencies and to maintain the situational awareness of the DR without having to depend on constant notifications. In Alpha, teams had access to a taskboard (via their smartphones) that showed predicted time to complete layers' recovery. Regarding the DR team's heightened level of visibility, the network operations center supervisor stated:

> *The advantages of this process were the ability to "login from anywhere" allowed me to view the near real-time status of the project from any device via Internet and the ability to provide feedback and ask questions in near real-time.*

The DR manager added:

> *[Teams] were able to focus on the critical items when they were able to see and to join the teams. So anything that required their attention would be put in that immediate context for them to focus on.*

Although the DR manager still received calls, texts, emails, and so on, the added visibility that the layered approach provides minimized the level of chaos and number of updates the DR manager received about the DR progress because Alpha's executives had a situational understanding about the recovery's status and next steps. With regard to taking a holistic approach to crisis response, Hale (1997, p. 238) argues that a layered model "enumerates all the functions necessary to support communication adequately while it disregards the details of final construction, leaving enough freedom for further design and construction". This capability helped teams to focus on the activities that pertained to a specific layer and left room for changes to the DR cadence. Thus, using the DR layers framework improved the visibility of the DR cadence over using runbooks.

Third, the DR layers framework promoted the organization to create and reuse DR documentation and maintain and update DR plans in an object-oriented fashion. Similar to the ISO model in which one creates reference architecture documents for each layer and detailed implementations follow a hierarchical documentation approach, the DR layers have a reference architecture that details the high-level elements that the documentation included in the sublayers that contain more operation documentation inherit. This layered approach allowed Alpha to create a non-traditional DR plan that was more modular (object oriented) in that one could align the plan maintenance responsibilities with the functional area that supports it. Where Alpha's traditional, task-level DR approaches were procedural, brittle, and contained a sequence of steps that detail the recovery activities for applications, the DR layers approach was flexible and easily combined with agile and lean frameworks to unify the DR efforts. The DR layer framework encourages one to breakdown DR documentation and artifacts and benefit from their reuse. By breaking down the documentation into layers and sublayers, the latter can inherit documentation from the former; in turn, the actual technology, processes, or people elements that compose DR can inherent the documentation from the sublayers, which encourages functional areas to reuse artifacts and aligns these artifacts with owners in the enterprise. The DR manager commented:

> *The DR layers framework promoted the creation and reuse of DR documentation by embedding resilience into the operation and inheriting into DR. [This approach is] changing the entire culture to create a culture of DR across all parts of the operation, where DR can inherit all the operational knowledge into a DR plan and present this knowledge to the right team at the right time during a recovery. The layers create a distributed approach to DR with each player understanding how they contribute to the whole, allowing a response with limited resources (as teams are also affected by disasters) that doesn't require command and control, but true orchestration.*

Table 4 compares the results of the traditional and layered approaches.

Table 4. Traditional vs. Layered Approaches to DR

Characteristic	Traditional	Layered
View of the DR work	Granular, task-level, application centric	Inheritance from framework to procedure
Emphasis	Documentation	Clear visualization
Documentation approach	Procedural / step by step	Object oriented and aligned with ownership and operational creation
Structure of the framework	Based on command and control, hierarchical	Distributed ownership with oversight
Modularity	Little to none, brittle	Promoted through layers and modularity
Key characteristics	Heavy documentation, focus on the plan	Necessary documentation, focus on value to the operation
	Focused on thoroughness of the documents for compliance	Focused on improving processes with multiple documents that together form the plan
	Void of a common structure to map the work to	Provide a common structure to map the work to
	Document provides common language for recovery teams	Framework provides common understanding to all stakeholders
	Promotes work completion	Promotes work business value
	Heavy documentation, focus on the plan	Necessary documentation, focus on value to the operation
Alignment	Supports a silo approach focused on applications or departments	Supports corporate alignment with focus on recovery priorities as they unfold
Communication with the business	Clear language for DR teams but vague or no common language for business	Clear language for all, the framework provides a view of the team's focus point and progress at all times

5 Conclusion

The proposed disaster recovery (DR) layer framework provides a useful yet simple framework for characterizing DR that includes the planning of its functions, its stages of recovery, and, ultimately, a unified framework for creating and understanding DR value. Outside of the findings we discuss above, we reflect on what we learned during this study and highlight this study's contributions as follows.

First, this research contributes to theory by highlighting the benefits of a layered approach to thinking about DR for enterprise architecture. When comparing the DR layers framework with other business continuity management (BCM) frameworks in the literature, the DR layers framework better accommodates the changing faces of DR and provides a better mental model for conceptualizing DR activities across a large enterprise. Prior BCM frameworks differ in structure (e.g., input-process-output models, staircase models, and custom frameworks) and scope (e.g., describing elements of BCM phases, providing a mechanism that is suitable to explain BCP to senior management, and focusing on the resiliency of computing technologies). In terms of structure, the DR layers framework leverages the strength of the staircase model (especially its adaptability and simplicity). In terms of scope, the DR layers framework spans both business continuity management (BCM) and business continuity planning (BCP) and fits well with more current DR thinking, which challenges the classic command and control (C2) assumptions (as Table 4 describes). The framework does not focus on the traditional components of BCP (i.e., developing, testing, and updating a plan) or the technologies that reduce downtime but rather on a way to map business continuity plans to a framework that can help one visualize and organize the recovery across the enterprise. Thus, this framework spans both the planning (preparedness) and recovery phases (McLoughlin, 1985) and considers the conceptualization of the recovery activities and the technologies involved in the recovery. Not only does such a conceptualization add to the BCP and DR literature, but it also yields important contributions for the DR practice. In Alpha (our case study organization), we found that the DR layers framework functioned as a link between the planning and recovery phases because the framework could provide a way to: 1) conceptualize related tasks by layer,

2) map these tasks to enterprise architecture (EA), 3) visualize the dependencies of the technologies, and 4) provide situational awareness when the initial plans broke down. As the recovery began, we found that the DR layers framework in conjunction with the taskboard provided 1) a better understanding and a better way to organize the DR tasks in the project backlog and 2) improved visibility concerning real-time changes that occurred during the recovery. Overall, the layers provide an abstraction mechanism for understanding the DR practice across an enterprise with a common framework. Importantly, the DR layers framework more clearly expresses the dependencies between IS and infrastructure.

Second, this research contributes to practice by testing the DR layers framework in an actual DR event. Some may argue that the uses of industry standards such as ITIL for creating DR plans are important in addressing disasters, but our findings suggest that the effectiveness of these industry-based DR plans were marginal during an actual disaster scenario when activities do not go according to plan. For instance, the normal "notification" channels often fail in situations with a significant loss of infrastructure. In fact, a recent disaster caused AT&T to lose all of its communication services for days, and communication was intermittent days after being restored (Gibbs, 2016). Alpha, as with any large company in healthcare, must have a sound DR plan with an aligned business continuity plan; however, lessons learned from disasters (Donahue & Tuohy, 2006) show us over and over how the planning structures used as best practices from which DR is born break. Our layered approach creates a common backdrop for understanding DR. By creating resilience and clear understanding of the elements around the DR layers framework, teams can rally and maintain focus on the recovery. Regardless of the conditions, because team members can join the recovery effort, they have the benefit of situational awareness during a dynamic and changing recovery. This approach enhances any ITIL-based communications plan.

While disaster plans provided Alpha with some value, they needed to break down the presence of plans themselves, make them visible to participants beyond the task level, make sure that a broad range of stakeholders could easily understand them, and make them able to accommodate changes. We found that many industry-standard models adopt command and control (C2) thinking and assume a high-level of operational maturity. Instead, Alpha refocused itself to gain value from building distributed robustness through a culture of prevention and resiliency, which provided value to the day-to-day operations. DR is now the result of operational readiness, where the organization embeds resilience into all of its activities. The approach to DR practice we describe in this study is less prescriptive than traditional BCP frameworks and more adaptable to multiple levels of operational maturity. Additionally, our framework accommodates lean and agile-based project management methods by providing benefits such as improving the organization and conceptualizing the project backlog.

Based on prior layered models, our DR layers framework is intuitive and easy to understand for non-technical stakeholders involved in DR. The framework is widely applicable and offers several other benefits to DR teams such as its focus on value, modularity, reuse and sharing of resources, unified understanding of the DR practice to support decision making, and corporate alignment. Overall, these benefits provide a better mental model for both DR and business stakeholders to approach DR activities. We hope researchers and practitioners alike can use our framework to gain insight on future studies or projects that involve DR activities.

Of course, our framework does have limitations. For instance, we tested our framework at a single company, Alpha. Future research needs to test the applicability of the framework in other situations because the recommended DR sublayers are unique to each different practice and implementation.

Through our collective common understanding of DR, researchers and practitioners will be able to develop a common industry framework that will help organizations form benchmarks for DR and gain a broader understanding of it.

References

Bajgoric, N. (2014). Business continuity management: A systemic framework for implementation. *Kybernetes*, *43*(2), 156-177.

Barnes, J. (2001). *A guide to business continuity planning*. New York: John Wiley.

Barroso, L., Dean, J., & Holzle, U. (2003). Web search for a planet: The Google cluster architecture. IEEE *Micro, 23*(2), 22-28.

Beamon, B., & Kotleba, S. (2006). Inventory modelling for complex emergencies in humanitarian relief operations. *International Journal of Logistics: Research and Applications*, *9*(1), 1-18.

Berke, P. R., Kartez, J., & Wenger, D. (1993). Recovery after disaster: Achieving sustainable development, mitigation and equity. *Disasters*, *17*(2), 93-109.

Brooks, C., Bedernjak, M., Juran, I., & Merryman, J. (2002). *Disaster recovery strategies with tivoli storage management*. IBM.

Bryson, K., Millar, H., Joseph, A., & Mobolurin, A. (2002). Using formal MS/OR modeling to support disaster recovery planning. *European Journal of Operational Research*, *141*(3), 679-688.

Chen, R., Sharman, R., Rao, H., & Upadhyaya, S. (2013). Data model development for fire related extreme events: An activity theory approach. *MIS Quarterly, 37*(1), 125-147.

Donahue, A., & Tuohy, R. (2006). Lessons we don't learn a study of the lessons of disasters, why we repeat them, and how we can learn them. *Homeland Security Affairs, 2*(2), 1-28.

Dubé, L., & Paré, G. (2003). Rigor in information systems positivist case research: Current practices, trends, and recommendations. *MIS Quarterly*, *27*(4), 597-636.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, *14*(4), 532-550.

Fallara, P. (2003). Disaster recovery planning. *IEEE Potentials*, *22*(5), 42-44.

Gibb, F., & Buchanan, S. (2006). A framework for business continuity management. *International Journal of Information Management*, *26*(2), 128-141.

Gibbs, C. (2016). AT&T suffers major network outage Saturday. *FierceWireless.* Retrieved from http://www.fiercewireless.com/wireless/at-t-suffers-major-network-outage-saturday

Hale, J. (1997). A layered communication architecture for the support of crisis response. *Journal of Management Information Systems, 14*(1), 235-255.

Harris, C. (2010). IT downtime costs $26.5 billion in lost revenue. *Information Week*. Retrieved from http://www.informationweek.com/it-downtime-costs-$265-billion-in-lost-revenue/d/d-id/1097919?

Herbane, B. (2010). The evolution of business continuity management: A historical review of practices and drivers. *Business History*, *52*(6), 978-1002.

Herbane, B., Elliott, D., & Swartz, E. (2004). Business continuity management: Time for a strategic role? *Long Range Planning*, *37*(5), 435-457.

Holdman, J. M., Kavuri, R., Rassbach, N., Hammett, S., & Ward, G. (2007). *Virtual tape storage system having snapshot virtual tape library for disaster recovery testing: Google Patents.* U.S. Patent No. 7,302,540.

Ireson, N. (2009). Local community situational awareness during an emergency. In *Proceedings of the 3rd IEEE International Conference on Digital Ecosystems and Technologies* (pp. 49-54).

IOS/IEO Commission. (1994). *ISO 7498-1: Information technology-open systems interconnection—basic reference model: The basic model.*

Jain, B. N., & Agrawala, A. K. (1993). *Open system interconnection: Its architecture and protocols.* New York: McGraw-Hill.

Jorgensen, D. (1989). *Participant observation*. New York: John Wiley & Sons.

Kappelman, L., Mclean, E., Luftman, J., & Johnson, V. (2013). Key issues of IT organizations and their leadership: The 2013 SIM IT trends study. *MIS Quarterly Executive*, *12*(4), 227-240.

Kendall, K., Kendall, J., & Lee, K. (2005). Understanding disaster recovery planning through a theatre metaphor: Rehearsing for a show that might never open. *Communications of the Association for Information Systems, 16*, 1001-1012.

Lawler, C., Szygenda, S., & Thornton, M. (2007). Techniques for disaster tolerant information technology systems. In *Proceedings of the 1st Annual IEEE Systems Conference* (pp. 1-6).

Lindström, J., Samuelsson, S., & Hägerfors, A. (2010). Business continuity planning methodology. *Disaster Prevention and Management*, *19*(2), 243-255.

McAvoy, J., & Butler, T. (2007). The impact of the Abilene Paradox on double-loop learning in an agile team. *Information and Software Technology*, *49*(6), 552-563.

McEntire, D., & Fuller, C. (2002). The need for a holistic theoretical approach: An examination from the El Niño disasters in Peru. *Disaster Prevention and Management*, *11*(2), 128-140.

McEntire, D., Fuller, C., Johnston, C., & Weber, R. (2002). A comparison of disaster paradigms: The search for a holistic policy guide. *Public Administration Review*, *62*(3), 267-281.

McLoughlin, D. (1985). A framework for integrated emergency management. *Public Administration Review*, *45*, 165-172.

Najafi, M., Eshghi, K., de Leeuw, S. (2014). A dynamic dispatching and routing model to plan/re-plan logistics activities in response to an earthquake. *OR Spectrum*, *36*(2), 323-356.

Nielsen, M. (2015). *The Google technology stack.* Retrieved from http://michaelnielsen.org/blog/lecture-course-the-google-technology-stack/

Pidd, M., de Silva, F., & Eglese, R. (1996). A simulation model for emergency evacuation. *European Journal of Operational Research*, *90*(3), 413-419.

Rubin, C. (1991). Recovery from disaster. In In T. E. Drabek & G. J. Hoetmer (Eds.), *Emergency management: Principles and practice for local government* (pp. 224-259). Washington, DC: International City Management Association,

Sandhu, R. J. (2002). *Disaster recovery planning*. Cincinnati, OH: Premier Press.

Sonnenwald, D., & Pierce, L. (2000). Information behavior in dynamic group work contexts: Interwoven situational awareness, dense social networks and contested collaboration in command and control. *Information Processing and Management*, *36*(3), 461-479.

Stein, S. (2004). *8th layer initiative*. North Carolina State University.

TAB Service Company. (2015). *Mississippi makeup business still recovering from data loss following Hurricane Katrina*. Retrieved from http://www.tabservice.com/records-management/mississippi-makeup-business-still-recovering-from-data-loss-following-hurricane-katrina/

Theoharidou, M., Kotzanikolaou, P., & Gritzalis, D. (2010). A multi-layer criticality assessment methodology based on interdependencies. *Computers and Security*, *29*(6), 643-658.

Department of the Army. (2011). *Operations* (FM 3-0). Washington, DC. Retrieved from http://fas.org/irp/doddir/army/fm3-0.pdf

Vescoukis, V., Doulamis, N., & Karagiorgou, S. (2012). A service oriented architecture for decision support systems in environmental crisis management. *Future Generation Computer Systems, 28*(3), 593-604.

von Steuben, F. (1779). *Regulations for the order and discipline of the troops of the United States part I.* Philadelphia: Styner and Cist.

Weick, K. (1993). The collapse of sensemaking in organizations: The Mann Gulch disaster. *Administrative Science Quarterly, 38*, 628-652.

Wester, J. (n.d.). What is kanban? *Everyday Kanban.* Retrieved from http://www.everydaykanban.com/what-is-kanban/

Yin, R. K. (2003). *Qualitative research from start to finish*. New York: Guilford.

About the Authors

Corey Baham is an Assistant Professor of Information Systems in the Spears School of Business at Oklahoma State University. His current research focuses on sociocultural factors that influence IS process agility. His work has been recently presented at major IS conferences, AMCIS and ICIS conferences, and is currently under submission at premier IS journals.

Andres Calderon has twenty-five years of diversified technology experience at both large and small enterprises; technical personnel management capability at different levels of the organizational structure; project management, and enterprise system administration experience. He also has vast experience in aligning technology to corporate vision and extensive background in business development.

Rudy Hirschheim is the Ourso Family Distinguished Professor of Information Systems in the E.J. Ourso College of Business, Louisiana State University. He has previously been on the faculties of the University of Houston (Houston, TX), McMaster University (Hamilton, Ontario), the London School of Economics (University of London), and Templeton College (University of Oxford). He was named AIS Fellow by the Association for Information Systems in December 2007 and given the LEO Award for Lifetime Achievement by the AIS in December 2013. He is senior editor for the journal *Information and Organization* and on the editorial boards of the journals *Information Systems Journal*, *Journal of Strategic Information Systems, Journal of Management Information Systems, Journal of Information Technology, and Strategic Outsourcing*. He has previously been on the editorial boards of the *European Journal of Information Systems,* the *Journal of the Association for Information Systems,* and *MIS Quarterly*.