HOUTKIN
CONSULTING, INC
TECHNICAL PROJECT MANAGEMENT
PROCESS ENGINEERING

WHITE PAPER

Disaster Recovery Testing

The information contained in this document is provided as a service and although we try to provide quality information, we make no claims, promises or guarantees, and assume no legal liability or responsibility, for the accuracy, completeness or adequacy of the information, products, or processes contained in the document or in any linked internet sites.

www.houtkinconsulting.com

Disaster Recovery Testing

Introduction.

Disaster recovery testing in the Corporate Private Sector is a science all until itself. Unlike application-level testing or singular infrastructure testing it includes testing from every angle: disaster recovery process testing; fail-over process and methodology testing; infrastructure-level testing; application-level testing; complete data center fail-over testing; end-to-end business process testing, and alternate seating site testing (with the alternate Data Center, if applicable).

Disaster Recovery testing teaches us to look at our business process and technical process and environments in as many views as possible to ensure thoroughness of deployment yet viability through its readiness. Disaster Recovery testing must ensure that there is no dependency on any facet of production and is the only method to test the viability of the resilience of any business.

Why Perform Disaster Recovery Testing.[1]

Because we just do not know if the Disaster Recovery environment works and whether we can actually fail-over to the Disaster Recovery environment and if we can, how long it will take and how many resources we need to invoke, maintain and normalize.

We make no assumptions in Disaster Recovery work. If it works in production, we cannot assume that it works in the disaster recovery environment. Why? Basically, we have to prove that the Disaster Recovery environment is viable, well-constructed, properly deployed and can fulfill the role it was meant for: supporting resilience of the Business when normal production business and technical venus are not available.

The testing process must prove its viability. It must prove that at any moment, not only does the complete disaster recovery environment work, that it also supports the fail-over process and meets the RTO. This means - no last minute configuration work, no last minute upgrades, no last minute patching. Disaster Recovery testing tests READINESS - not our ability to perform break-fix under pressure. Once an incident occurs, it is already too late. Staff cannot simultaneously focus on their safety and which variable they left the servers in because they did not have a chance to “get to it” because of some other work.

Here are some other reasons why we perform Disaster Recovery testing.

1. Disaster Recovery testing provides an opportunity to test the complete Disaster Recovery process.

   The Disaster Recovery process begins once the Incident Management Team invokes the Disaster recovery process. Testing provides an opportunity to test all of these processes at least once a year resulting in muscle memory for everyone in the Business - to be referred to in case of an incident.

2. Software Life-Cycle Testing does not test the Disaster Recovery Environment.

   Application-level testing performed during the software development life-cycle process focusses directly on the application and its functionality without reference to the complete environment that supports the end-to-end business process. Testing is performed in the QA or Integration test environment, not the the Disaster Recovery environment so that application in Disaster Recovery is not exercised.

3. Use of the Application/System/Technology in the Disaster Recovery Environment on a daily basis.

   The Business tends to use the application deployed in the production environment. Infrastructure Teams focus on their kit and related services deployed in production to ensure optimum business support.

   Unless you have so designed your environment, the Disaster Recovery environment is not exercised on a regular basis to ensure that all of the little things that are fixed on a daily basis in production actually work actually work in the Disaster Recovery environment.

4. Change Management Process may not include Disaster Recovery environment.

[1] Please note that this refers to a phased-in approach to technical testing and business process testing of the disaster recovery environment that is built to manifest the requirements delineated in the Business Continuity plans. This is not Business Continuity testing which comprises: evacuation drills; Staffing Accountability process testing; Call Tree testing; Walkthrough of the Business Continuity Plan or individual staff/business visits to the alternate seating site that is performed during a normal business day; e.g. connectivity to the production network.

As soon as the application is deployed on production kit and integrated into the production environment, it is used on a regular basis with issues being responded to via the Change Management process and phase 2 development cycle, if required. In some companies, the Change Management process does not include the Disaster Recovery environment and so it is not always clear whether Changes made to better the production environment are made in the Disaster Recovery environment.

5. Deployment schedule of Production and Disaster Recovery.

   Applications deployed in production are not always deployed simultaneously or even within 24 hours in the Disaster Recovery environment. As well, changes to the production environment as a result of the Change Management process may not be directly made to the Disaster Recovery environment and may linger on a "punch-list" of things to do when time permits.

6. Differences between the network and hardware from both Production and Disaster Recovery environment.

   The Disaster Recovery network is different from the production network. Disaster Recovery kit is a different from the production kit. Just because an application server's network configuration is correct and that the servers works in production, we cannot assume that a different server used in Disaster Recovery actually works or that its network port configuration is correct.

7. Disaster Recovery Testing tests the complete business process, not just individual applications.

   We approach Disaster Recovery testing with a phased-in methodology - making testing more and more complex as we prove the viability of the environment. The Business tests its process on a daily basis even with the integration of a new application, system or technology. Since the Disaster Recovery is not used on a regular basis, application or core-level testing is not enough to prove that the Disaster Recovery environment is viable or that the Business will actually be able to work in times of an incident.

8. Disaster Recovery Applications are Deployed Using the Same Configurations as Production.

   If the Disaster Recovery environment is deployed/configured as it is in the production environment, then the Disaster Recovery applications and infrastructure will be trying to find servers in production impacting the fail-over time-line that could impact the Business RTO. The Disaster Recovery environment should be configured to support the scenario with the highest risk - usually loss of the primary/production data center.

9. Disaster Recovery testing tests the fail-over process and methodology. There is no other kind of testing where this work is done.

   The Disaster Recovery testing environment is the only environment where the full fail-over process and methodology can be tested.

10. Disaster Recovery testing tests a completely different Data Center along with a completely different Seating Environment. Only Disaster Recovery testing performs this kind of full environmental testing.
11. Disaster Recovery testing tests the full fail-over of one Data Center to another Data Center.
12. Disaster Recovery testing tests the Disaster Recovery Scenario that is identified in the Business and Facility Impact Analysis and Risk Assessment.

A Note Regarding Scenarios.

The Disaster Recovery scenario is the backbone to planning for both the Business and the Disaster Recovery Teams. Each scenario has its own nuances but there are only three (3) basic scenarios and two technical options that is - which technical environment the Business continues to access - whether from within the primary facility or from an alternate seating site or remotely, from home or a hotel. Simply put, you either have to leave the building, you cannot get into the building, or you must stay in the building.

Keep it simple and the planning and testing scope will fall into place depending on the scenario. Always remember, though, when testing, you test the worst case scenario, which means you test your disaster recovery environment. This is the point of the testing.

Here are the generic scenarios.

Scenarios	Impacted	Technical Environment	
Scenario 1: Impact to Facility	Data Center	Disaster Recovery	Business Seating is still maintained in the Facility Technology determines whether to resolve immediate Data Center problem or fail-over to the Disaster Recovery environment
	Seating	Production	Business moves to alternate seating environment or from home, accessing the network via remote access. The Data Center maintains status quo so staff are still connecting to the production technical environment but from remote seating.
Scenario 2: Denial of Physical Access	Data Center	1. Production 2. Disaster Recovery	Business moves to alternate seating environment or from home, accessing the network via remote access. Depending on the impact of the Data Center, the Data Center can continue running and be remotely managed to a point - until maintenance is required. However, this is the path of least resistance until the Business and Technology Management know whether they need to fail-over to the Disaster Recovery Environment. So, at the beginning, the Business would be connected to the production environment. If failover to the Disaster Recovery environment is required, the Business would be interrupted while the fail-over takes place.
	Seating	Production	Business moves to alternate seating environment or from home, accessing the network via remote access.
Scenario 3: Shelter-in-Place	Building Lockdown	Production	Here, staff continues working from their normal seating environment and connecting to the production technical environment -- unless they were outside of the facility when the lock-down occur - in which case they can work from home or from the alternate seating site - depending on the plan. Depending on the situation within the building and the direction of the Fire/Safety Director, staff may not be able to actually go to the Data Center floor or work from their seats so there may be a period of time when Management will have to wait to see how the situation unfolds before they choose to fail-over to Disaster Recovery technical environment.

A Philosophical View to Disaster Recovery Testing.

Because disaster recovery testing takes the complete business process and solutions in mind, one can wax poetic regarding the academic and philosophical ideas about disaster recovery as a discipline. The reader should note that the author is neither an academic nor a philosopher but a feet-on-the ground technician who has experienced over 6 disasters. With this experience and even more experienced Colleagues, we have created various ways to test the Disaster Recovery environment.

In a nutshell, we have to determine:

1. Is the core infrastructure (hardware) maintained as it is maintained in production? As technology is upgraded in production, is it also upgraded in disaster recovery?
2. Are Change Management processes and procedures in place for the Disaster Recovery environment?

3. Are patching schedules for production and disaster recovery in synch?
4. Are applications upgraded in disaster recovery on the same schedule as in production?
5. Is the disaster recovery environment “dr ready?” Is the disaster recovery infrastructure and application front-ends configured to support the scenario with the highest risk?
6. Is the network (router access lists/firewall) configuration and change management process in working order for the disaster recovery environment?
7. Is operational process and configuration documentation up-to-date?
8. Is the fail-over methodology a viable solution? Is the technical fail-over process viable?Does everyone know what the process is? Has cross-training in the technical environment occurred? Are roles/responsibilities identified for technical staff?
9. Does the replication methodology meet the Business requirement and RPO?
10. Does the replication process work? Are there data integrity problems? Issues with the fail-over process to the alternate data repository? How often is the disaster recovery repository tested?
11. Has the complete business process been documented and manifested in disaster recovery as in production? Is core infrastructure, circuits, protocols in place in disaster recovery as they are in the production environment?
12. Does the alternate seating site meet current business requirements?
13. Is the same service paradigm in production available in the Disaster Recovery environment?
14. Is there a communications process in place with all Third-Party Business and Vendor Partners?
15. Is there a testing program in place with all Third-Party Business and Vendor Partners?

Simply put, disaster recovery testing tells us:

1. Whether Management and the Business support the Business Continuity and Disaster Recovery program; e.g. do we have Business participation in the test?; Is there enough resources available to keep the Disaster Recovery environment production-ready?; Are staff cross-trained?; Is the Disaster Recovery and Business Continuity environments properly documented - the documentation maintained as the Business changes?; Is the technical team overwhelmed with Management requests for production support that they cannot keep the Disaster Recovery environment maintained?
2. Whether the Business has identified a disaster recovery mission and RTO that can be met by the Technology team;
3. Whether the CIO supports disaster recovery by ensuring that production process and support embrace the disaster recovery environment and that the decision of architectural standards; technical standards and products all take disaster recovery into consideration so that disaster recovery is not an after-thought but part and parcel of production.
4. Whether the Business Continuity team and approach are viable and whether there is good communication with the Disaster recovery team who manifest the requirements into solutions.
5. Whether the technology team has fully thought-out the implications of their disaster recovery process.
6. Whether there is a viable technical fail-over process and whether this process is supported by database and development teams through best practices.

So, we need to create a testing program that touches upon all aspects of the Disaster Recovery process, solutions, operational procedures as well as the Business Continuity planning process and Risk Assessment.

The more that companies integrate disaster recovery life-cycle into the production process, the faster disaster recovery precepts will be accepted as a part of the production process, with the same decision regarding resource allocation and vision for Technology.

How Issues Become Visible During Testing.

There is no great secret to creating a Disaster Recovery testing program. What does dissuade many people, though, is the amount of work that is required.

- It is not unusual to plan a test only to find out that a staging server has not been deployed because no-one looked at all of the components supporting a particular business process or even took the time to document the end-to-end business process and map to the current production components to see what is required for deployment in disaster recovery.
- Many focus on replication methodologies without considering how they manage to perform the manual process of

failing from one repository to another and the time and resources required.

- Some fail-over the front-end and the databases, bring them up in the disaster recovery environment only to find that the sql.ini, tnsnames.ora or the odbc files cannot find the databases deployed in the disaster recovery environment.
- Business staff show-up at the alternate site to find that there is no seat for them, their workstation is not working, they cannot be authenticated, there are issues connecting to the core network, one-off applications are not fully deployed or do not work; fax machines do not work, and/or full authentication is not available to required databases.
- Applications cannot be invoked because they are dependent on a file that is installed on another server that was not identified as part of the core components for the application so it is not deployed in Disaster Recovery.
- Technical Staff or Business Staff cannot become authenticated to an application or database only to find that application-level/database-level permissioning was not responded to during disaster recovery deployment - or it was assumed that it was replicated. Replication does not copy files from root in some cases.
- And last, but not least, one of my favorites, configuration changes, software upgrades and hardware upgrades are not responded to in disaster recovery - leading to an interrupted testing process and sometimes weeks to perform problem resolution to determine that software notes were not read; that there were no change management processes supporting the disaster recovery environment or there were so many production projects placed on the backs of the technology team that they could not focus on disaster recovery.

All-in-all, technology staff do not realize the amount of time it takes to trouble-shoot the small problems while they are also dealing with their own response to an incident; the lack of full communication with everyone; working in a data center you are not familiar with; not being able to find the Engineer who was responsible for this server, etc.

This is the net/net of Disaster Recovery Testing. If it is not there for testing, it is not there for an incident. It's too late.

Disaster Recovery is not a Break-Fix Process.

We do not assume that we can just "fix-it" during an incident because of the variables:

- Staff dis-engage from the process and go home and do not communicate for days. Or worse, people die and we do not know for several weeks. So, the excuses I hear no longer work:
  1. "I'll just re-configure it "on the fly" in DR" What if you cannot make it to the site? What if you left your laptop in the building and the building is damaged?
  2. "Sam will take care of it. That's his expertise." What if you cannot find Sam and his expertise handled a mission critical service or system?
  3. "DR is about how quickly and how much you can perform in a short period of time." What if you are injured, scared, can't get to the site, can't get a remote connection. What if your family needs you and you cannot focus on the disaster recovery site?
- Communications may be out.
- Travel may be non-existent.

The realities about our current-day incidents are that they can be catastrophic. There is no longer any excuse for not focussing on READINESS. That is what we test in Disaster Recovery testing. We do not test our abilities to be heroes. Because, the truth is that we really do not know what will happen and what the variables will be when an incident occurs.

Disaster Recovery planning and readiness is about being responsible to the Business that you work for. It is not about individuals and their capabilities. To focus on immediate fixes means that the technical solution is a kluge and we cannot be sure that the business can survive on a kluge.

The Components of the Testing Program.

The components that comprise the testing program are:

- Management Support
- Business Management Support
- Technical Test Facilitator
- Business Continuity Representative(s)
- The Test Plan
- Testing Scripts by Test Type/Application/Technology/Infrastructure
- Logistics
- The Command Center
- The Testing Participants
- The Testing Proctors
- The Team

Management Support.

Management support greases the wheels for the testing vehicle. It supports the point behind testing; e.g. validation as well as eases issues revolving around resource allocation and is an approval mechanism for the test date.

Business Management Support.

Business Management is our "owner" of the business process and so they help in identifying test participants and taking a role in understanding where there are issues and how we can best respond to them.

Technical Test Facilitator.

Disaster Recovery testing requires someone who understands process from all levels and can simultaneously see individual facet of the test while keeping the complete business and technical environments in mind.

Business Continuity Representative.

Business Continuity representatives for each business whose assets are being tested or who are participating in the test should participate into the process to ensure the viability of the business continuity planning, business impact analysis, risk assessment processes.

The Test Plan.

The Disaster Recovery Test Plan is not a test script. The Disaster Recovery Test Plan documents:

- the complete test planning process;
- identifies the disaster recovery scenario that is being tested[2],
- the scope of the test (what business process, technology, technology process, hardware and software that is involved in testing);
- test methodology (use of an isolated network; simulation of certain processes to avoid impact to production;
- test objectives that provide the metrics for measuring success of the test and overall management sign-off. data entry for those scenarios where the dependent system is not-in-scope, etc;
- the test chronology;
- test logistics;

[2] The Disaster Recovery Scenarios are an important benchmark in identifying the scope of the Test. The scenarios, in brief, are: Scenario 1: Facility is Impacted While Inhabited. Here we have two sub-scenarios: Loss of Data Center or Production Computing Facility; 2) Full Loss; Scenario 2: Denial of Physical Access; Scenario 3: Shelter-in-Place.

- Issues listing with follow-up and root-cause analysis/date of remediation
- Re-testing, if required,
- Post mortem/final test results
- All Management sign-off

Test Scripts

Since the testing approach is a phased-in one, each phase should have its own test script. So, we have a test script for:

Test Type	Test Plan/Test Script	Comments
Installation Validation Testing (IVP)	IVP Test Script Only	NA
Technical Testing (TT)	TT Test Script Attached IVP of Tested Infrastructure	NA
Application Functionality Testing (AFT1) - Core Infrastructure Only	• Test Plan • AFT Test Script for Each Application	NA
Application Functionality Testing (AFT2) - Including Up-/Down-Stream Dependencies	• Test Plan • AFT Test Script for Each Application	1. The test plan explains the full technical scope. 2. The same application-level test script as used in AFT1 may be used in AFT2
End-User Test	• Test Plan • AFT Test Script for Each Application	1. The test plan explains the full technical scope. 2. The same application-level test script as used in AFT1 or AFT2
Site Fail-Over Test	• Test Plan • AFT Test Script for Each Application • AFT Test Script for Each Technology	1. The test plan explains the full technical scope. 2. The same application-level test script as used in AFT1 may be used in AFT2

Logistics.

You will need to ensure that the alternate seating area can support the Test Participants and the Command Center. There should be parking, regular availability of food and coffee and other refreshments. All directions and travel instructions should be available in the Test Plan.

More than not staff are not being paid extra to perform this testing. They should be so honored with good food and a pleasant working experience so that their complete feeling about the Disaster Recovery solution is a positive one.

Many of the technical teams will work in the Data Centers. They should also be fed and taken care of and they should be constantly connected to the Command Center so phones/conference phones are a must in the Data Center area.

A board will be required to post issues or notifications.

Physical security process should be maintained during the testing - and staff need to be reminded that they are responsible to the complete Disaster Recovery process and ensure that they have the proper identification with them.

Any area that is supporting Test Participants should have ample stationery, copiers and fax machines, if required for the test. They should all be available and ready for use.

All logistical information should be advertised at least 1 month prior to the test.

The Command Center.

The Command Center is the brain and life-force of an incident. It should always be manned once the Disaster Recovery process is invoked by the Incident Management Team. The Disaster Recovery test is a great time to really exercise the Command Center Process. The existence of the Command Center teaches teams that there is a central point for status, for information and for problem resolution.

It is requested that Technology Managers always be connected to the Command Center Bridge and gather status of their teams during the failover and invocation period. It is important, however, to bring issues into the Command Center so that the various technical disciplines can work to resolve them. This fosters a sense of team that can be carried forth, with muscle-memory, in times of an event.

The Command Center should follow a strict process of what is discussed on the bridge; what should become an off-line conversation; an ability to focus on a strict schedule of status for Staff and Management; a focus on problem resolution.

The Testing Participants.

Everyone is engaged in the Disaster Recovery testing process

1. The Business - to test the environment, the process and their applications
2. Management - to perform the Declaration process, call-in for Status and provide support.
3. Technology - to test their process, cross-training and readiness
4. Business Continuity Team - to test the Business plans, the Business Continuity process
5. Disaster Recovery Team - to test their process
6. Business/Vendor Partners - to keep focussed on the success of their technical and Business services during testing

The Testing Proctors.

Everyone has a tendency to become creative during testing.

Because the Disaster Recovery environment is not reviewed on a regular basis, it is important that the level of detail is responded to. If there is an issue, we need to understand the process that fostered the issue; the application, the variable. Every piece of information is important in performing trouble-shooting. And, while we are trouble-shooting during the period after the test and before the post-mortem, we still have an injured Disaster Recovery environment until the root cause is identified and the resolution applied and the process/technology/system is re-tested.

Proctors help us to ensure that Business and Technical test participants keep to the test script so that we can better refine the problem resolution process in a timely fashion.

The Team.

We always try to approach Disaster Recovery testing as a team-building event. It is imperative that everyone see the importance of the Team - especially during an incident. Disaster Recovery testing needs to simulate an incident - and it is the value and benefits of working as a team that really helps the Business and Staff make it through an incident.

The Testing Inventory.

Because we have many facets that we need to look at, we have many levels of testing. The testing inventory includes hardware-level check-off and testing as well as the testing of system software, infrastructure services, utilities, middleware (if applicable) and the use of business applications to provide the infrastructure change through the complete network and component-level environment.

There are several sub-levels of testing in the system/application/technology/service testing program. Each process or test is designed to test different facets of the application: infrastructure only; infrastructure and core application components; baseline application functionality; application functionality in relation to other system dependencies; business process and the application of a member of a larger infrastructure-related environment.

The pre-dominate testing types include:

- Infrastructure Verification Sign-Off
- Technical Testing
- Individual System/Application/Technology- Core Infrastructure
- Individual System/Application/Technology Functionality Test with all core components and those components comprising up-/down-stream dependencies
- End-to-End Process Testing: Single Process.
- End-to-End Process Testing: Multiples Processes
- End-User Testing
- Site Fail-Over Testing

Infrastructure Validation Process	Whenever a new piece of infrastructure is installed in or deployed to the disaster recovery environment	Going forward, to be re-visited when changes are made to the infrastructure whether on a hardware or software level
Technical Testing	Whenever a new application/system or service is deployed. Performed after the IVP	Going forward, to be re-visited when changes are made to the application/system software
Application Functionality Testing-1	Whenever a new application/system or service is deployed. Performed after the TT	Whenever new functionality or an upgrade occurs on the application-level
Application Functionality Testing-2	Whenever a new application/system or service is deployed. Performed after the AFT1	Whenever a new up-/down-stream dependency is added to an existing application
End-User Testing	Whenever a new application/system or service is deployed. Performed after AFT2	Revisited at the annual data center site failover testing
End-to-End Business Process: Single/Multiple	Annual or as identified by the Business	
Site Fail-Over: Technical	Annual with a 3-month planning period	Some companies perform this on a bi-annual basis based on the status of their data centers
Site Fail-Over: Full	Annual with a 3-month planning period	Some companies perform this on a bi-annual basis

Infrastructure Verification Process (IVP).

Also known as the Infrastructure Validation Process, this test is the approval point for the deployment of the core infrastructure of an application/system/technology or service. Here, the infrastructure owner(s) approves of the deployment by performing a check-out and approval of the configuration (swap space, processing speed, memory, storage allocation and mapping) in respect to:

- the sizing of the application/database functionality based on Business requirements for the business processes supported by the application/system/technology/service
- mainframe and distributed services such as: operating system (OS) and related operating system services: dhcp, dns, ddns, wins, etc.
- addressing, partitioning, printing queues, connectivity to the network and useable protocols: ping, echo or trace-route)
- architecture verification: active/active, active/passive
- front-end to back-end connectivity, baseline OS-related functionality between servers)
- core functionality of the software on the application-level (database, front-end (web/application) executables

When to Perform IVP.

- Upon deployment of a hardware device supporting the DR deployment of an application/system or infrastructure-level technology.

Who Performs the IVP.

- The Infrastructure Engineers who performed the deployment.
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script.

What is tested.

- Device configuration (physical)
- Operating System Installation and configuration (software)
- Operating system version patching and hardening
- Network Card configuration and connectivity to the network
- Drive configuration
- Special services configuration to support the application/technology/system
- Middleware: Cognos, IIS, etc.
- Replication software deployment, methodology and physical-level configuration
- SAN device installation and mapping configuration

Guidelines.

- The server must be isolated from the production network during deployment and IVP
- All application/database development-related upgrades, changes must not be made during testing. Testing is not an opportunity to perform this work but rather an opportunity to test these changes scheduled and implemented through the Change Management process.

Technical Testing (TT).

Also known as “shakedown”, this process tests the application/system/technology from the perspective of its functionality after the specified application/system/technology service is installed on top of the already validated hardware and OS/services (via the IVP). Technical Testing is the testing of both the core infrastructure and application-levels.

When to Perform TT.

- After deployment and successful IVP
- After deployment of the application-level software and/or database

Who Performs the TT.

- Application Developer/Manager/Owner,
- Infrastructure Staff provides support
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script.

What is tested.

- Testing the relationship of the core components that comprise an application
- Operating system configuration (software) changes from that in the IVP
- Operating system version patching and hardening compatibility with application/database
- Drive configuration changes from that documented and checked in the IVP
- Special services configuration to support the application/technology/system
- Application installation and configuration
- Database instance installation and configuration
- Database installation and configuration
- Replication software deployment and replication methodology and configuration
- SAN device installation and mapping configuration

Guidelines.

- If data is replicated, it is important to gather Business-approval for stopping replication during the period of time that the TT is performed. The Business needs to understand the implication of stopping replication to their work.
- All issues are to be resolved and TT performed again before moving to the Application Functionality Testing
- Data entry is not permitted in TT-level testing
- The Application and Database Server(s) must be isolated from the production network during testing
- All application/database development-related upgrades, changes must not be made during testing. Testing is not an opportunity to perform this work, but rather an opportunity to test these changes schedule and implemented through the Change Management process
- The Application front-end should be pointing to the correct database to support the testing that supports the worst-case Disaster Recovery Scenario[3]. Each configuration set should be documented in the Application Invocation Document for each application and technology
- After testing, if data is SAN-based, the disaster recovery storage should be de-imported from the SAN and replication re-engaged
- After testing, if data is local-based, the data replication configuration should be quickly reviewed prior to re-engaging the replication process
- The application/database software and related hardware must be returned to its original architectural integrity after testing

[3] Disaster Recovery Testing is always performed to “prove” a particular scenario. There are no assumptions in Disaster Recovery Testing so each scenario, if applicable, must be tested. The scenarios are: Scenario 1: Unsafe Facility; Scenario 2: Denial of Physical Access; 3: Shelter-in-Place

- the disaster recovery application should be left pointing to the configuration that supports the riskiest Scenario for the business to minimize failover in case of an incident
- All issues are to be resolved and proper testing performed again before moving to Application Functionality Testing. If there is a problem on the infrastructure/OS level, both IVP and TT must be re-visited before moving to Application Functionality Testing. If the issues is related to the application and/or database-level only, TT must be revisited

Application Functionality Testing 1-(AFT1): Core Infrastructure.

This process tests baseline functionality of the system/application/technology/service using the core infrastructure.

When to Perform AFT1.

- After deployment of a successful TT for the infrastructure and software-level deployments and TT

Who Performs the AFT1.

- Application Developer
- Infrastructure Staff provides support
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script

What is tested.

- Basic functionality of the application/database
- Data Integrity through the comparison of disaster recovery reports with that from the production system
- Ability to the invoke the application from the disaster recovery network
- Logon capability to the application/system/technology/service via the front-end or client
- Scroll-through of the application to ensure that all functions are available
- Use of existing data to test reporting functions
- Controlled data entry to test those functions that require new data

Guidelines.

- If data is replicated, it is important to gather Business-approval for stopping replication during the period of time that the TT is performed. The Business needs to understand the implication of stopping replication to their work.
- All issues are to be resolved and TT performed again before moving to the Application Functionality Testing
- Data entry is encourage in AFT1-level testing. Only the use of test data is permitted. The use of production-related data to resolve production-related business process is not permitted. This is non-negotiable. AFTG1 testing is not an opportunity to perform production-related work.
- All application/database develop-related upgrades, changes must not be made during testing. Testing is not an opportunity to perform this work, but rather an opportunity to test changes changes schedule and implemented through the Change Management process.
- The application and Database Server(s) must be isolated from the production network during testing.
- The Application front-end should be pointing to the correct database to support the testing that supports the worst-case Disaster Recovery Scenario. Each configuration set should be documented in the Application Invocation Document for each application and technology.
- After testing, if data is SAN-based, the disaster recovery storage should be de-imported from the SAN and replication re-engaged
- After testing, if data is local-based, the data replication configuration should be quickly reviewed prior to re-engaging the replication process.
- The application/database software and related hardware must be returned to its original architectural integrity after testing.
- the disaster recovery application should be left pointing to the configuration that supports the riskiest Scenario for the business to minimize failover in case of an incident.
- All issues are to be resolved and proper testing performed again before moving to Application Functionality Testing-2. If there is a problem on the infrastructure/OS level, both IVP and TT must be re-visited before moving to Application Functionality Testing-2. If the issues is related to the application and/or database-level only, TT must be revisited.

Application Functionality Testing 2-(AFT2): Core Infrastructure and Up-/Down-Stream Dependencies.

This process tests baseline functionality of the system/application/technology/service using the core infrastructure.

When to Perform AFT2.

- After deployment of a successful AFT1 test and/or after the deployment of a new up-/down-stream dependency

Who Performs the AFT2.

- Application Developer
- Infrastructure Staff provides support
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script
- Representative of the up-/down-stream dependency

What is tested.

- Everything performed in AFT1
- Application functionality utilizing up-/down-stream dependencies

Guidelines.

- As identified for the AFT1

End-User Testing.

This testing gleans approval of the End User for the complete deployment of an application or business process, in the disaster recovery environment. This test permits the Business to perform Business-As-Usual testing of those individual enterprise-wide and business-related applications/systems/technology services the prove out any changes, revisions to the existing or newly deployed infrastructure levels of the enterprise including: power, network, device-level: mainframe, infrastructure components supporting OS services and enterprise-wide application services including: file share utilities/servers, clustered database environment: web services, application platform environments: authentication systems and ftp gateway for communication with Trading Partners as well as application/database support infrastructure that may have been upgraded in operating system or operating system configuration.

When to Perform End User Testing.

- After the AFT2 for a new application or new infrastructure, functionality, dependencies to an existing mission critical application
- During the annual site failover test
- Based on the request of the Business Owner

Who Performs the End User Testing.

- Business Owner or Business Representative
- Desktop Engineering
- Application Developer
- Infrastructure Staff
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script
- Representative of the up-/down-stream dependency

What is tested.

- Everything performed in AFT2
- Application functionality utilizing up-/down-stream dependencies
- Possibly additional applications to support the End User Request
- The disaster recovery desktop or the production desktop (to test the scenario where User Seating has not changed, but the computing facility has failed to the disaster recovery environment.)

Guidelines.

- As identified for the AFT2
- The same test scripts utilized for AFT1 and AFT2

Business Process Testing: Single/Multiple.

This testing gleans tests a single or multiple business process(es) that would be required to be performed during an incident, if failover to the disaster recovery environment is invoked. Here, there is review of multiple systems, databases, dependencies, Users and workstations. This type of testing is an imperative in the Disaster Recovery testing toolbox but it is often not performed because the Disaster Recovery environment requires too much attention from a deployment perspective - or there are too many issues during component-level testing that the team never makes it to testing end-to-end process.

When to Perform End-to-End Business Process Testing.

- After the AFT2 for a new application or new infrastructure, functionality, dependencies to an existing mission critical application
- During the annual site failover test
- Based on the request of the Business Owner

Who Performs the End User Testing.

- Business Owner or Business Representative
- Desktop Engineering
- Application Developer
- Infrastructure Staff
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script
- Representative of the up-/down-stream dependency

What is tested.

- Everything performed in AFT2
- Those applications, systems and dependencies that comprise the end-to-end business process being tested
- The disaster recovery desktop or the production desktop (to test the scenario where User Seating has not changed, but the computing facility has failed to the disaster recovery environment.

Guidelines.

- As identified for the AFT2

Site Failover Testing: Data Center.

The Data Center site failover test is the true test of the Disaster Recovery environment primarily because the site is on its own without accessibility to the production site. Here, everything is tested - from the physical side of the Data Center up through to the application-level. It provides us an opportunity to identify the validity of the declaration process, fail-over process and methodology as well as whether the Business RTO is reachable. As well, Site Failover testing provides the team with the chance to engage the business in the complete process.

When to Perform Site Failover Testing.

- At least once a year usually scheduled
- After a major change to the Data Center or a move to a new Data Center

Who Performs the End User Testing.

- Business Owner or Business Representative
- Network Engineering
- Server Infrastructure engineering
- Desktop Engineering
- Help Desk Support
- Application Development Team
- Representative of the up-/down-stream dependency
- The Business Continuity Team
- The Disaster Recovery Test Facilitator provides over-sight of the testing process and ensures adherence to the test script

What is tested.

Infrastructure

- The alternate data center physical environment: power/environmentals/design/labelling
- All network routers- edge; distributed and core including access lists, routing, protocols
- All firewalls and firewall policy configuration
- All infrastructure servers and services: dns, dhcp, wins, domain controllers, license, vm, management systems
- All ftp and staging servers supporting up-/down-stream dependencies
- All middleware servers and services
- All application servers
- The disaster recovery desktop or the production desktop (to test the scenario where User Seating has not changed, but the computing facility has failed to the disaster recovery environment.

Applications

All applications, supporting the mission critical and critical business process, required to be viable in the Disaster Recovery environment

Processes

- Declaration process
- The Command Center Process
- Technical failover methodology: dns re-direct, for example
- Technical failover process:
    - Fail-over chronology
    - Stopping replication
    - Re-zoning of storage
    - Promotion of the disaster recovery domain controllers: authentication to the network
    - Running of the disaster recovery logon scripts
    - Any infrastructure-level authentication; e.g. Active Directory, if applicable
    - Start-up of database processes
    - Start-up of databases
    - Start-up of Infrastructure Utilities, Citrix, Job Scheduling
    - Start-up of Messaging Services
    - Start-up of Middleware servers
    - Startup of Application Server processes
- Final Check-out of the Infrastructure Environment and hand-off to the application development team.
- Final Check-out of the applications and hand-off to the business
- Engagement of the Help Desk Process
- Specific Business Process
- Desktop Engineering Process
- Test Process
- Normalization Process: Roll-back to Production
- Post Mortem Process

Guidelines.

- Data Center site failover testing is highly visible. Approval must be gained by the CEO, the CIO and all Business Heads regarding the process, the preparedness schedule and the actual date of performance.
- A consistent communication program is required to communicate with Senior Management and Staff to remind them of the event and their responsibilities.
- A consistent reporting program is required to keep the Business posted and to keep in-touch with the Business in case their plans change and the test requires re-scheduling.
- Communication with all Third Party Business and Vendor Partners where isolation of the production data center could potentially cause impact to their service or the send/receipt of files
- Communication with the Data Center (Operations), Facilities and Corporate Security is mandatory.
- A 3-month planning period.
- A Testing Team comprised of:
    - Representatives from all of the business is required.
    - Representatives from all of technical disciplines is required.

    NOTE: It must be made very clear that everyone testing must attend the weekly test meeting.

- In order for the test to make sense to the community, the following items must be communicated with every notification to remind the Community as to what the testing is about and its meaning to their business.
  - The Disaster Recovery Scenario being tested
  - The implication of the Disaster Recovery scenario
  - The building in scope for testing
  - Testing Methodology: isolation of the testing environment to diminish risk to the production environment
  - Access to the production network and what services will or will not be available for those who need to/want to work on the weekend of the test
    - Access to group/home/personal shares
    - Access to email
    - Access to applications
    - Blackberry/Other PDA
    - Market Data Services
    - Custodial Systems and Files and/or up-/down-stream dependencies
    - Internet via the Core Network (for remote access Users)
  - The complete begin and end-time of the test. Testing fail-over begins on a Friday at roughly midnight (depending on the Company's batch process) and continues until Sunday at 2:00 PM (depending on their global business requirements).
  - Businesses participating in the test
  - The Test Plan must document everything for Compliance
  - You may need to ensure that Internal Audit is available for testing and that they have a contact throughout the test to answer questions
  - The Testing process must be air-tight and follows it's own protocol:
    - Weekly Meetings with a strict Agenda: All meetings must start and end on-time.
    - Off-Line meetings to manage deliverables and issues is required
    - Strict Deliverable Schedule: Test Scripts
    - Clear Delineation of Roles/Responsibilities
    - Detailed chronology of the testing process and actions delineated by Date/Time and Roles/Responsibilities. This becomes the facilitator's guideline to the complete test and helps to identify failover times on an application level to set metrics for review and benchmarking for each successive test
    - Clear Message that all who are testing must attend the test meetings
    - There is always 3 meetings the week of the test to review deliverables of the week and the testing chronology and deal with last-minute issues.
    - The day-of-test is documented on the Chronology
    - All Test Scripts are gathered at the end of the test
    - The post mortem section of the Test Plan must be completed before the post mortem meeting. It may take up to 2-weeks to perform the post mortem
    - The post mortem meeting may have follow-ups with are to be managed.
    - Once all of the issues are resolved, the test script is finalized and a final report is distributed.

Summary.

Hopefully, after reading this detailed approach to Disaster Recovery testing, you can see the importance of performing this work. The primary reason we perform Disaster Recovery testing is to mitigate the risk that often is not documented in the Business Continuity Plan: that of non-attention to the actual solutions and logistics that have been manifested in order to support the business plan. We cannot mitigate risk to the Business if we do not take care of the solution that we deployed to mitigate risk in the first place.