OT Recovery Readiness

A Practical 30–60–90 Day Improvement Plan

Macrium Software

Most organizations believe they can recover critical systems after an outage or cyber incident. Far fewer have actually proven it.

This 30–60–90 day framework provides a structured approach for OT, IT, and security teams to strengthen recovery readiness in manufacturing environments.

The plan helps teams:

- Identify the systems that matter most to production
- Validate whether recovery objectives are realistic
- Strengthen backup architecture and recovery processes
- Prove recovery capability through drills

The Recovery Readiness Journey

Recovery readiness typically improves in four phases

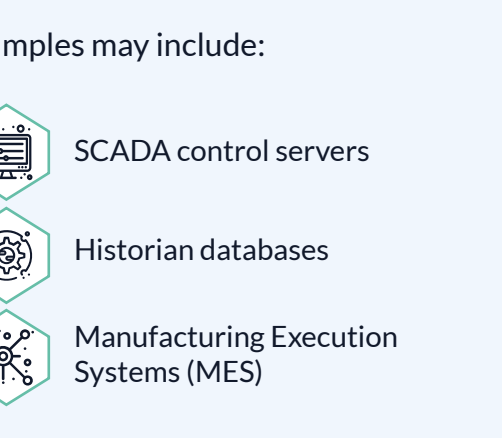
DEFINE → ARCHITECT → VALIDATE → OPERATIONALIZE

Phase	Timeline	Focus	Outcome
DEFINE	Days 1–30	Identify systems, dependencies, and recovery objectives	Clear visibility into recovery priorities and constraints
ARCHITECT	Days 30–60	Clear visibility into recovery priorities and constraints	Architecture supports real-world recovery scenarios
VALIDATE	Days 30–60	Test restores and measure recovery performance	Proven recovery capability and identified gaps
OPERATIONALIZE	Days 60–90	Formalize processes, runbooks, and drills	Formalize processes, runbooks, and drills

This phased approach ensures organizations:

- Define what needs to be recovered and in what order
- Design infrastructure that supports real-world recovery
- Validate recovery capability through testing
- Operationalize recovery so it is repeatable and reliable

Recovery readiness is not achieved through backup deployment alone. It is achieved when systems can be restored, validated, and operated under real-world conditions.



Phase 1 — Days 1–30

Define Your Recovery Requirements

Objective
Define the systems, dependencies, and recovery requirements that matter most to production.
Many organizations discover recovery gaps simply because system dependencies and recovery procedures have never been documented.

Step 1: Identify Critical Production Systems

Critical production systems are the applications or infrastructure components that would stop production if they were unavailable.

Examples may include:

- SCADA control servers
- Historian databases
- Manufacturing Execution Systems (MES)
- Engineering workstations
- Identity services
- Virtualization hosts

Document ownership and recovery objectives for each system.

Step 2: Map System Dependencies

Critical systems rarely operate independently. Production environments typically rely on multiple layers of supporting infrastructure.

Example Dependency Stack

- Production Line
- SCADA Control System
- Historian Database
- SQL Infrastructure
- Virtualization Hosts
- Active Directory
- Network Infrastructure

Understanding these relationships allows teams to define the correct recovery order.

Restoring systems without understanding dependencies can significantly delay recovery.

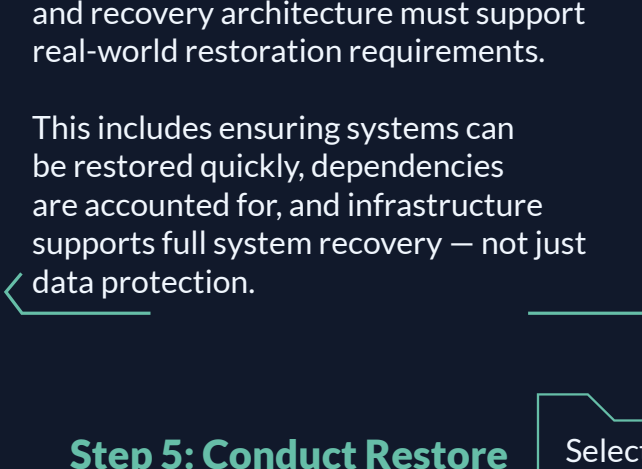
Step 3: Review Backup Coverage

Evaluate whether existing backup infrastructure supports full system recovery.

Key questions to answer:

- Are system-level backups available for critical infrastructure?
- Are backup copies stored outside the production security domain?
- Are immutable or offline backup copies available?
- Are backups verified regularly?

This step establishes a baseline view of the organization's current recovery capability.



Phase 2 — Days 30–60

Architect & Validate Recovery Capability

Objective
Test whether documented recovery objectives can actually be achieved under realistic conditions.
Many organizations discover at this stage that their recovery assumptions differ significantly from real-world restore performance.

Step 4: Architect for Recovery

Before recovery can be validated, backup and recovery architecture must support real-world restoration requirements.

This includes ensuring systems can be restored quickly, dependencies are accounted for, and infrastructure supports full system recovery — not just data protection.

Step 5: Conduct Restore Testing

Select several high-priority systems and perform controlled restore tests. Testing may include:

- Restoring system images into isolated environments
- Verifying operating system startup
- Confirming application functionality

Critical System Inventory (An example of what to include)

System	Business Owner	Technical Owner	RTO	RPO	Last Restore Test

The goal is to prove that systems can actually be restored successfully. To learn how to build a resilient backup validation strategy you can rely on, read our OT Engineer's Playbook for Backup Validation.

Step 6: Measure Recovery Performance

Capture metrics during restore testing, including:

1. Restore throughput
2. Time required to restore infrastructure
3. Application startup time

Key Metric: Time-to-Validated-Restore (TVR)
Time-to-Validated-Restore (TVR) measures the time from the start of a recovery event until:

- Systems are restored
- Applications are operational
- Data integrity is verified
- Business users confirm the system works

Tracking TVR provides a practical measure of recovery readiness. Many organizations monitor backup job success. TVR measures something more important: how long it takes to restore real business capability.

Step 7: Identify Recovery Bottlenecks

Restore testing often reveals operational constraints such as:

- Infrastructure limitations
- Undocumented dependencies
- Manual recovery steps
- Authentication dependencies

Document findings and assign owners to address gaps.

Phase 3 — Days 60–90

Operationalize Recovery Capability

Objective
Strengthen backup architecture and formalize recovery processes so recovery becomes repeatable and reliable.

Step 8: Improve Backup Architecture

Address gaps discovered during testing. This may include:

- Implementing system-level imaging
- Improving storage resilience
- Introducing immutable or offline backups

Backup architecture should support rapid system restoration, not just data retention.

Step 9: Document Recovery Procedures

Create recovery runbooks that include:

- System dependencies
- Recovery order
- Restore procedures
- Validation steps

Documentation ensures recovery processes do not rely on tribal knowledge.

Step 10: Conduct a Recovery Drill

Simulate a realistic outage scenario involving multiple systems.
Example drill structure:

1. Restore infrastructure services
2. Restore application systems
3. Validate integrations
4. Verify business workflows

Recovery Drill Template

Scenario	Systems Involved	Recovery Order	Target RTO	Actual TVR

Recovery drills provide the most reliable validation of recovery readiness.

What Success Looks Like After 90 Days

Organizations completing this program should have:

- Documented crown-jewel systems
- Mapped system dependencies
- Validated restore capability
- Measured recovery performance
- Established recovery runbooks
- Conducted at least one recovery drill

Key Takeaway

Backups provide the foundation for resilience. Recovery readiness is proven only when systems can be restored and validated under real-world conditions.
If a system has never been restored and tested, its recovery capability remains uncertain.