Technology Domain Guide

Build the digital foundation that makes intelligent work practical, safe, and reliable.

Document Version 1.1 FINAL | TIO-KR-2026.08.0 | August 2026

Table of Contents

How to use this guide

Use the opening sections for the domain view. Use the capability sections to understand definitions, boundaries, outputs, evidence, measures, and maturity. Use the final sections to support assessment, planning, and implementation.

01 Purpose and Use of This Master

This document captures the working definition of the TIO Technology Capability Domain and its five constituent capabilities.

This is a master definition document. It should support framework writing, capability assessment, statement of work design, workshop design, maturity scoring, roadmap development, service design, and future translation into assessment and planning tools.

02 TIO Technology Domain

Technology is the organization's ability to provide the digital foundation that allows intelligent work to happen safely, practically, and reliably.

In TIO, Technology is not only information technology support and it is not only AI tooling. It is the enterprise capability to design, select, connect, automate, protect, and improve the technology environment that the whole organization depends on.

Technology turns enterprise direction into usable systems. It gives Data a place to operate, Factory a place to build, People a set of tools they can actually use, and Leadership a foundation it can trust.

Domain Role

- Provide the technology foundation for intelligent work across the enterprise.
- Translate enterprise strategy into practical technology choices and roadmaps.
- Make platforms usable, owned, connected, governed, and continuously improved.
- Connect systems, data, workflows, AI services, and automation tools safely.
- Enable automation without losing accountability, reliability, or trust.
- Protect critical technology services so the organization can keep operating when systems fail, threats appear, or vendors change.

Domain Operating Logic

Architecture shapes. Platforms enable. Integration connects. Automation improves. Resilience protects.

Domain Principles

- Technology serves enterprise direction. It does not define direction by itself.
- Technology choices must make work easier, safer, faster, more reliable, or more valuable.
- AI tools are part of the technology estate, not a separate side environment.
- Architecture should reduce confusion and technical debt, not create a document nobody uses.
- Platforms must have owners, standards, support models, and improvement plans.
- Integration should remove repeated manual effort and make governed data flow across work.
- Automation must stay connected to human accountability, evidence, and value.
- Resilience is not only cybersecurity. It includes reliability, recovery, continuity, monitoring, vendor risk, and trust.
- Technology must be understandable enough for leaders to govern and practical enough for teams to use.
- Each capability must be evidenced, measured, improved, and buildable.

The Five Technology Capabilities

Capability	What it does	Plain-English outcome
Architecture	Designs the shape of the enterprise technology foundation.	Leaders and teams know what technology environment they are building toward.
Platforms	Selects, configures, governs, and improves major systems and tools.	The organization gets more value from the systems it already owns and uses.
Integration	Connects systems, data, workflows, and services.	Work moves across boundaries without constant rekeying, manual joins, and status chasing.
Automation	Uses technology and AI-enabled workflows to perform repeatable work safely and measurably.	Routine work is redesigned and improved without losing human accountability.
Resilience	Keeps technology services secure, reliable, recoverable, compliant, and trusted.	The organization can trust and recover the systems that intelligent work depends on.

2026.08 ALIGNMENT | Technology in the refreshed TIO system

Primary outcome contribution: Intelligence, Automation, and Enterprise Autonomy

- Technology gives the organization approved intelligence access, model and platform choice, integration, deterministic and AI-enabled automation, identity, observability, resilience, and credible exit paths.
- A single logical enterprise experience may use multiple replaceable models, tools, services, and human specialists.
- Portability, recovery, and provider change are designed capabilities rather than contract slogans.

Evidence at managed maturity: Approved platforms, integration, automation, identity, portability, observability, resilience, and exit capability are measured.

03 Standard Capability Definition Template

Each TIO capability should use the same structure, so the full framework stays consistent and easy to translate into tools, assessments, workshops, and statements of work.

Required Capability Structure

- Capability title: one-word name wherever possible.
- Short definition: what the organization must be able to do repeatedly.
- Plain-English Test: a simple question a senior leader can understand.
- Description: plain-English explanation of why it matters and how it works.
- Why it matters: the client problem this capability solves.
- Five behaviors: the repeatable actions that make the capability real.
- What this capability owns and does not own: boundary protection.
- What people actually do: the day-to-day activities and roles that make the capability real.
- Real-world example: a practical example without client identifiers.
- Practical outputs: no more than three concrete deliverables, routines, or artifacts.
- Evidence of maturity, useful measures, maturity model, owners, service connection, and boundary notes.
- SOW Test, top-down validation, and bottom-up validation.

Practicality Standard and SOW Test

- Every TIO capability must be simple to name, clear to define, practical to build, observable in operation, measurable in maturity, and useful enough to scope into real work.
- Every capability must pass the SOW Test: Could a client hire someone to help build this capability, and would both parties understand the expected outputs?
- Every capability must produce no more than three practical outputs. This keeps the framework useful, memorable, and deliverable.
- Every capability must include What People Actually Do and a Real-World Example so non-technology executives can see how the capability works in practice.
- Each capability must be validated from the top down and from the bottom up before it is treated as stable.

04 Architecture

Short Definition: Architecture is the organization's ability to design, maintain, and adapt a coherent technology foundation that connects systems, data, AI, automation, workflow, security, and governance.

Plain-English Test: Can leaders, technology teams, and business owners explain the current technology environment, the target direction, the major gaps, the key standards, and how technology choices support the enterprise strategy?

Description

Architecture helps the organization decide what technology environment it is building toward and how the pieces should fit together. It gives structure to systems, platforms, data flows, automation, AI tools, security controls, vendor choices, and governance requirements.

A mature Architecture capability does not stop at a diagram. It gives leaders a practical way to make technology decisions, reduce unnecessary complexity, avoid tool sprawl, manage dependencies, and sequence investment. It translates enterprise direction into a technology foundation the organization can actually build and use.

Why It Matters

- Without Architecture, technology decisions become local, reactive, vendor-led, or tool-led.
- Architecture creates a shared view of the current state, target state, standards, dependencies, and trade-offs.
- It helps leaders decide what to connect, what to modernize, what to retire, what to protect, and what to defer.

Five Behaviors

Behavior	Meaning
Map	Create and maintain a clear view of systems, data flows, integrations, technology owners, and major dependencies.
Design	Define the target-state technology foundation needed to support strategy, data, AI, automation, and work redesign.
Choose	Set practical architecture principles that guide build, buy, partner, integrate, secure, and retire decisions.
Sequence	Turn the target architecture into a realistic roadmap based on value, risk, cost, readiness, and capacity.
Adapt	Update architecture as strategy, technology, risk, workforce needs, and evidence change.

What This Capability Owns

- Current-state and target-state technology architecture.
- Architecture principles and standards.
- Major technology dependencies and sequencing logic.
- Build, buy, partner, integrate, and retire decision support.
- The technical roadmap that supports enterprise direction.

What This Capability Does Not Own

- Enterprise strategy itself.
- Detailed data governance rules.
- All platform administration activities.
- Every project plan or backlog item.
- Governance decision rights, which sit in Leadership and are deployed through Technology where needed.

What People Actually Do

People make this capability real by turning technology ideas into repeated decisions, working routines, and operating evidence.

Role	What they actually do
Executive Sponsor	Confirms the business direction, investment posture, risk appetite, and major technology choices.
CIO, CTO, or Technology Lead	Owns the architecture discipline, prepares options, manages standards, and explains trade-offs.
Enterprise or Solution Architect	Maps systems, designs target patterns, documents dependencies, and tests options.
Business and Process Owners	Explain work reality, priorities, system pain points, and operational constraints.
Security, Risk, Data, and Privacy Leads	Confirm that architecture choices can support control, trust, evidence, and compliance.

Simple activity flow

1. Clarify the enterprise direction and technology problem.
2. Map the current systems, data flows, integrations, constraints, and owners.
3. Define the target technology foundation and architecture principles.
4. Identify gaps, risks, dependencies, and sequencing choices.
5. Review, fund, and adapt the roadmap based on evidence and changing conditions.

Real-World Example

A growing service organization has customer records in one system, field work in another, finance in a third, and reporting in spreadsheets. Teams spend time chasing status and reconciling data instead of serving customers.

The architecture work creates a current-state map, defines a target foundation, and sets principles for which systems will be authoritative, which integrations matter first, and which platforms should not be expanded. In real life, Architecture shows up as fewer random tool decisions, clearer investment choices, and a roadmap people can actually follow.

Practical Outputs

Output	What it contains
Technology Landscape Map	A practical map of major systems, owners, data flows, integrations, dependencies, risks, and pain points.
Target Architecture	A plain-English target-state view showing how systems, data, AI, automation, workflow, security, and governance should fit together.
Architecture Roadmap	A sequenced plan of technology changes, decisions, dependencies, and investment choices needed to move from current state to target state.

Evidence of Maturity

- Current-state technology landscape map.
- Target architecture documented and approved.
- Architecture principles used in real decisions.
- Named owners for major systems and technology domains.
- Roadmap linked to enterprise strategy and priority AI Use Cases.
- Decision log showing why major technology choices were made.
- Evidence that redundant, risky, or outdated technology is being managed.

Useful Measures

Measure	What it tells us
Architecture clarity	Percentage of leaders and delivery teams who can explain the target direction and key standards.
Decision alignment	Percentage of major technology decisions reviewed against architecture principles.
Technical debt reduction	Reduction in redundant tools, unsupported systems, manual workarounds, or avoidable dependencies.

Maturity Model

Level	Maturity Description
0 Absent	No meaningful technology architecture exists. Systems, tools, and integrations are understood only locally.

Level	Maturity Description
1 Initial / Ad-Hoc	Some architecture knowledge exists, but it depends on individuals and is not used consistently in decisions.
2 Repeatable	The organization can map major systems and discuss target direction, but standards and roadmaps are uneven.
3 Defined	Architecture is documented, owned, communicated, and used in major technology and AI decisions.
4 Managed	Architecture is tied to funding, risk, roadmap decisions, technical debt management, and regular review.
5 Optimized	Architecture continuously adapts as strategy, technology, data, risk, and work design change.

Typical Owners and Participants

Role	Typical participants
Executive sponsor	CEO, COO, CIO, CTO, business unit executive, city manager, deputy city manager, or equivalent.
Accountable owners	CIO, CTO, enterprise architect, technology lead, transformation lead.
Participants	Business owners, data leaders, security leaders, privacy leaders, risk leaders, operations leaders, finance, procurement, and delivery teams.

Domain Boundary Notes

- Architecture owns the shape and sequencing of the technology foundation.
- Platforms owns the operation and improvement of major systems and tools.
- Data owns data quality, stewardship, lineage, and trust.
- Leadership owns strategic direction, investment posture, governance design, and risk appetite.

SOW Test

PASS: A TIO user/client could hire support to map the current technology landscape, define a target architecture, create architecture principles, and build a practical roadmap. The expected outputs are clear and buildable.

Top-Down Validation

Executives can see whether technology investment is aligned to enterprise direction, whether major risks and dependencies are understood, and whether choices are being made deliberately.

Bottom-Up Validation

Teams can see which systems matter, which standards apply, which pain points are being addressed, and how architecture decisions affect their daily work.

05 Platforms

Short Definition: Platforms is the organization's ability to select, configure, govern, support, and continuously improve the major systems and tools people use to get work done.

Plain-English Test: Do people know which platforms to use for which work, who owns them, how they are configured, what rules apply, and how the organization will improve or retire them?

Description

Platforms turns software from a collection of licenses into a managed business asset. It covers the major systems people depend on: enterprise systems, productivity suites, collaboration tools, workflow tools, AI platforms, analytics tools, customer systems, finance systems, sector systems, and shared operating tools.

A mature Platforms capability is not measured by how many tools the organization owns. It is measured by whether platforms are useful, governed, adopted, integrated, secure, supported, and improved. The goal is to help people work better with the systems already in place before buying more.

Why It Matters

- Without Platforms, the organization pays for tools that are underused, duplicated, poorly configured, or locally owned without enterprise discipline.
- Platforms creates ownership, standards, support, adoption, configuration discipline, and improvement routines.
- It helps leaders get value from technology investments instead of confusing activity with capability.

Five Behaviors

Behavior	Meaning
Select	Choose platforms based on enterprise need, fit, risk, cost, interoperability, usability, and long-term value.
Configure	Set up platforms so they support real work, clear roles, security, data needs, workflow, reporting, and AI-enabled use where appropriate.
Own	Assign clear business and technology ownership for each major platform.
Improve	Review platform usage, pain points, adoption, cost, support issues, and improvement opportunities at a regular cadence.
Retire	Reduce tool sprawl by consolidating, replacing, or retiring platforms that no longer justify their cost, risk, or complexity.

What This Capability Owns

- Platform inventory and rationalization.
- Platform ownership and stewardship model.
- Configuration standards and improvement priorities.
- Platform adoption, support, and usage discipline.
- Platform lifecycle decisions in partnership with Architecture, Finance, Security, Procurement, and business owners.

What This Capability Does Not Own

- The target architecture by itself.
- Every user training activity.
- Detailed data stewardship.
- All vendor procurement decisions.
- All workflow redesign decisions that belong in Factory or operations.

What People Actually Do

People make this capability real by turning technology ideas into repeated decisions, working routines, and operating evidence.

Role	What they actually do
Platform Owner	Owns platform purpose, value, roadmap, business fit, and improvement priorities.
Technology Administrator	Configures, supports, secures, monitors, and maintains the platform.
Business Owner	Defines how the platform should support work, reporting, customer service, compliance, or operations.
Super Users or Champions	Help teams use the platform well, surface issues, and test improvements.
Procurement, Finance, Security, and Privacy	Review cost, contracts, risk, access, data handling, and vendor obligations.

Simple activity flow

1. Inventory the major platforms and tools in use.
2. Confirm what each platform is for, who owns it, and what business outcomes it supports.
3. Assess fit, adoption, configuration, cost, risk, overlap, and improvement needs.
4. Create a platform roadmap that improves, consolidates, integrates, or retires tools.
5. Review usage, value, support issues, and risks on a regular rhythm.

Real-World Example

A managed enterprise owns Microsoft 365, a customer system, a finance system, a workflow tool, and several department-specific applications. People still manage important work through email, spreadsheets, and unofficial trackers because the platforms are not configured around real workflows.

The Platforms capability clarifies ownership, improves configuration, reduces duplicate tools, and creates a roadmap for better use. In real life, Platforms shows up when people know where work should happen, when systems are easier to use, and when leaders stop buying new tools to compensate for poorly used existing ones.

Practical Outputs

Output	What it contains
Platform Inventory and Fit Assessment	A clear view of major platforms, owners, users, costs, purpose, adoption, risks, overlaps, and improvement opportunities.
Platform Ownership Model	Named business and technology owners, decision rights, support responsibilities, and review cadence for each major platform.

Output	What it contains
Platform Improvement Roadmap	A short practical plan to configure, integrate, consolidate, improve, or retire platforms based on value and risk.

Evidence of Maturity

- Approved platform inventory.
- Named platform owners.
- Usage, cost, and adoption data reviewed at cadence.
- Configuration standards and support model documented.
- Platform roadmap linked to business priorities and Technology architecture.
- Reduction in duplicate, unused, unsupported, or risky tools.
- User feedback incorporated into platform improvement decisions.

Useful Measures

Measure	What it tells us
Platform adoption	Usage and active adoption of strategic platforms by intended users and teams.
Platform value	Evidence that platforms reduce friction, improve work, support decisions, or enable priority AI Use Cases.
Tool rationalization	Number of redundant, unsupported, or low-value tools consolidated, improved, or retired.

Maturity Model

Level	Maturity Description
0 Absent	No reliable platform inventory exists. Tools are purchased, configured, and used locally with little enterprise visibility.
1 Initial / Ad-Hoc	Some major platforms have owners, but usage, configuration, support, and improvement are inconsistent.
2 Repeatable	The organization can list major platforms and owners, but value, overlap, adoption, and lifecycle decisions are uneven.
3 Defined	Platform ownership, configuration, support, usage expectations, and improvement routines are documented and communicated.
4 Managed	Platforms are measured for adoption, value, risk, cost, support quality, and lifecycle decisions.
5 Optimized	Platforms continuously improve, integrate, simplify work, and adapt as enterprise needs change.

Typical Owners and Participants

Role	Typical participants
Executive sponsor	CIO, COO, CFO, business unit executive, city manager, deputy city manager, or equivalent.
Accountable owners	Platform owners, CIO, CTO, application owners, business owners, product owners.
Participants	Technology administrators, business users, super users, data stewards, security, privacy, finance, procurement, and vendors.

Domain Boundary Notes

- Platforms owns the major systems and tools people use.
- Architecture owns the broader technology shape and standards.
- Integration owns how platforms connect to other systems and workflows.
- People owns broader workforce adoption, change, and training experience.

SOW Test

PASS: A TIO user/client could hire support to inventory platforms, assess fit and overlap, define ownership, improve configuration, and create a platform roadmap. The expected outputs are clear and practical.

Top-Down Validation

Executives can see whether the organization is getting value from its technology investments and whether major platforms are owned, governed, and improving.

Bottom-Up Validation

Employees can see which tools to use, where to do the work, how to get support, and which platform problems are being fixed.

06 Integration

Short Definition: Integration is the organization's ability to connect systems, data, workflows, and services so work can move across boundaries with less manual effort and more reliable information.

Plain-English Test: Can the organization explain which systems need to connect, what information should move, who owns the flow, what controls apply, and where integration would remove the most friction?

Description

Integration is what turns separate applications into a working technology environment. It connects systems, data, workflows, platforms, AI services, and operational processes so people do not have to act as the glue between systems.

A mature Integration capability does not connect everything for its own sake. It connects the flows that matter most for value, risk, service quality, automation, data trust, and user experience. It is practical, sequenced, governed, and tied to real work.

Why It Matters

- Without Integration, people rekey information, reconcile spreadsheets, chase status, copy data between systems, and make decisions from incomplete views.
- Integration creates reliable information flow across work, reduces manual effort, and enables better data, automation, reporting, and AI Use Cases.
- It helps the organization move from isolated tools to connected operations.

Five Behaviors

Behavior	Meaning
Trace	Identify where work and information enter, move, stall, duplicate, or break across systems and teams.
Prioritize	Choose integration needs based on value, risk, frequency, user pain, data readiness, and automation potential.
Design	Define what data or events move, when they move, where they move, who owns them, and what controls apply.
Build	Implement and test integrations using appropriate connectors, application programming interfaces, integration tools, workflows, or data pipelines.
Monitor	Track integration health, failures, exceptions, ownership, and improvement needs over time.

What This Capability Owns

- Integration map and priority backlog.
- Integration patterns, standards, and ownership.
- Data and workflow movement between systems.
- Integration health, exception handling, and monitoring.
- Technical connection points that enable workflow automation and AI-enabled work.

What This Capability Does Not Own

- Data definitions and data quality ownership.
- Full process redesign by itself.
- Platform selection by itself.
- Business decision rights.
- All reporting or analytics design.

What People Actually Do

People make this capability real by turning technology ideas into repeated decisions, working routines, and operating evidence.

Role	What they actually do
Process Owner	Explains the workflow, handoffs, pain points, exceptions, and business value of integration.
Integration Lead or Architect	Designs the connection pattern, data movement, ownership, and standards.
System Owners	Confirm what each system can send, receive, expose, protect, and log.
Data Steward	Validates meaning, quality, sensitivity, and source-of-truth issues.
Security and Privacy Leads	Review access, data handling, logging, consent, and risk controls.

Simple activity flow

1. Identify high-friction workflows where people move information manually.
2. Map source systems, target systems, data fields, events, owners, and controls.
3. Prioritize integrations based on value, risk, readiness, and effort.
4. Design, build, test, and document the integration.
5. Monitor failures, exceptions, data quality issues, and user experience after release.

Real-World Example

A customer request enters a portal, is manually copied into a service system, then copied again into a finance tracker and weekly reporting spreadsheet. Each handoff creates delay, errors, and status meetings.

The Integration capability connects the portal, service system, finance fields, and reporting flow. In real life, Integration shows up as fewer duplicate entries, fewer status emails, faster service, better reporting, and a stronger foundation for automation.

Practical Outputs

Output	What it contains
Integration Map	A map of the systems, data flows, handoffs, manual joins, ownership points, and integration gaps that matter most.
Priority Integration Backlog	A ranked list of integrations to build, improve, monitor, or defer based on value, risk, readiness, and effort.
Integration Standards	Practical rules for application programming interfaces, connectors, data movement, logging, ownership, error handling, and monitoring.

Evidence of Maturity

- Integration map exists and is kept current.
- Priority backlog is linked to business value and automation needs.
- Integration standards are used in project and platform decisions.
- System owners and data owners are identified for key flows.
- Integration failures and exceptions are monitored.
- Manual rekeying, duplicate entry, or reconciliation work is reduced.
- New AI Use Cases can access governed information without unsafe shortcuts.

Useful Measures

Measure	What it tells us
Manual handoff reduction	Reduction in rekeying, spreadsheet reconciliation, duplicate entry, and status chasing.
Integration reliability	Uptime, failure rate, exception volume, recovery time, and data movement accuracy.
Flow improvement	Cycle time, error rate, or user effort reduction in workflows supported by integration.

Maturity Model

Level	Maturity Description
0 Absent	Systems are largely disconnected. People move information manually and system relationships are poorly understood.
1 Initial / Ad-Hoc	Some integrations exist, but they are built one-off and depend on individual knowledge.
2 Repeatable	The organization has a basic way to identify and build common integrations, but standards and monitoring are uneven.
3 Defined	Integration map, backlog, standards, owners, and design patterns are documented and used consistently.
4 Managed	Integrations are measured, monitored, governed, and connected to workflow, automation, and data decisions.
5 Optimized	Integration continuously improves information flow, reduces friction, supports automation, and adapts as platforms and work change.

Typical Owners and Participants

Role	Typical participants
Executive sponsor	CIO, CTO, COO, operations executive, business unit executive, or equivalent.
Accountable owners	Integration lead, enterprise architect, platform owners, process owners, data owners.
Participants	System administrators, data stewards, security, privacy, business analysts, workflow leads, vendors, and frontline users.

Domain Boundary Notes

- Integration owns connection and flow between systems.
- Data owns quality, definitions, stewardship, lineage, and trust.
- Factory owns AI Use Case delivery and workflow redesign.
- Platforms owns the systems being connected.

SOW Test

PASS: A TIO user/client could hire support to map critical system flows, identify integration pain points, prioritize the backlog, define standards, and design first integrations. The outputs are concrete and actionable.

Top-Down Validation

Executives can see which disconnected systems are blocking value, what integrations matter most, and how integration reduces friction and enables automation.

Bottom-Up Validation

Teams can see fewer duplicate entries, fewer manual workarounds, fewer missing updates, and clearer ownership when data or workflow movement fails.

07 Automation

Short Definition: Automation is the organization's ability to use technology, workflow tools, and AI-enabled systems to perform repeatable work safely, measurably, and under human accountability.

Plain-English Test: Can the organization identify repeatable work, decide what should be automated, define human review, test the workflow, measure value, and stop or scale based on evidence?

Description

Automation turns the technology environment into operational leverage. It uses workflow tools, rules, scripts, robotic process automation, AI-assisted workflows, and AI agents where appropriate to reduce manual effort, improve speed, improve quality, and make work more consistent.

A mature Automation capability does not chase automation for its own sake. It starts with real work, clear value, human accountability, risk controls, measurement, and an Evidence Gate. It connects closely to Factory, which selects and delivers AI Use Cases, but Technology owns the automation environment, tooling, standards, controls, and reliable operation.

Why It Matters

- Without Automation, people keep performing repeatable tasks manually even when better tools exist.
- Automation creates capacity, consistency, speed, quality, and better use of human judgment.
- It helps the organization redesign work around what technology can do well and what people must still own.

Five Behaviors

Behavior	Meaning
Identify	Find repeatable tasks, rule-based work, high-volume handoffs, common requests, and workflow friction suitable for automation.
Design	Define the future workflow, human review points, exception rules, controls, metrics, and ownership.
Build	Configure or engineer the automation using approved platforms, integration patterns, AI tools, and security rules.
Measure	Compare performance to baseline using cycle time, error rate, effort, cost, quality, user experience, and risk measures.
Govern	Use evidence to decide whether the automation should continue, redesign, scale, pause, or stop.

What This Capability Owns

- Automation environment, tools, technical standards, and operating controls.
- Automation candidate backlog in partnership with Factory and business owners.
- Technical design, testing, monitoring, and support of automated workflows.
- Human review and exception routing design in partnership with Leadership, Factory, Risk, and People.
- Automation measurement support and operational reliability.

What This Capability Does Not Own

- Enterprise strategy and value priorities by itself.
- All AI Use Case selection and portfolio governance.

- Detailed role redesign and reskilling.
- Data readiness and data quality ownership.
- Risk appetite and governance design, which are owned by Leadership and applied through Technology.

What People Actually Do

People make this capability real by turning technology ideas into repeated decisions, working routines, and operating evidence.

Role	What they actually do
Business or Process Owner	Confirms the work problem, baseline, value measure, exceptions, and accountability.
Automation Lead	Designs and coordinates the automation pattern, build, test, release, monitoring, and support.
Technology Builder	Configures workflow tools, bots, connectors, scripts, or AI-enabled automation components.
Risk, Privacy, Security, and Legal Reviewers	Confirm boundaries, human review, access, records, privacy, and risk controls.
Users and Managers	Test whether the automation works in real work, identify exceptions, and confirm adoption needs.

Simple activity flow

1. Start with a real work problem and baseline.
2. Decide which tasks are suitable for automation and which require human judgment.
3. Design the future workflow, controls, review points, exception handling, and measures.
4. Build, test, release, and monitor the automation.
5. Use evidence to scale, redesign, continue, or stop.

Real-World Example

A finance team manually reviews vendor documents, extracts key fields, checks for missing information, and routes items for approval. The work is repetitive, but some exceptions require human judgment.

Automation extracts fields, checks completeness, flags exceptions, and prepares an approval queue. A human still approves the vendor decision. In real life, Automation shows up as faster processing, fewer errors, clearer exceptions, and people spending more time on judgment rather than administrative assembly.

Practical Outputs

Output	What it contains
Automation Candidate Backlog	A practical list of automation opportunities with purpose, owner, baseline, value measure, data needs, risk level, and readiness.
Automation Control Design	A workflow design showing tasks automated, human review points, exception handling, access controls, logging, and accountability.
Pilot Automation Scorecard	A simple evidence pack comparing baseline to results and recommending continue, redesign, scale, pause, or stop.

Evidence of Maturity

- Automation backlog exists and is prioritized.
- Automations have named owners and human accountability.
- Baselines and success measures are defined before build.
- Human review and exception paths are documented.
- Automations are tested before production use.
- Performance, failure, and user feedback are monitored.
- Evidence Gate decisions are made and documented.

Useful Measures

Measure	What it tells us
Capacity release	Hours, effort, or workload reduced or redirected after automation.
Quality and speed	Cycle time, error rate, rework, throughput, and service quality before and after automation.
Governed operation	Percentage of automations with owner, controls, monitoring, review rules, and Evidence Gate decisions.

Maturity Model

Level	Maturity Description
0 Absent	No structured automation capability exists. Automation is scattered, unmanaged, or avoided.
1 Initial / Ad-Hoc	Individuals or teams create local automations, but ownership, controls, value, and support are inconsistent.
2 Repeatable	The organization can identify and deliver basic automations, but measurement and governance are uneven.
3 Defined	Automation backlog, design standards, controls, owners, measurement, and review routines are documented and used.
4 Managed	Automation is measured, monitored, governed, improved, and connected to evidence-based scale, redesign, or stop decisions.
5 Optimized	Automation continuously improves work, releases capacity, supports new operating models, and adapts as technology and work change.

Typical Owners and Participants

Role	Typical participants
Executive sponsor	COO, CIO, CTO, CFO, transformation sponsor, business unit executive, or equivalent.
Accountable owners	Automation lead, process owner, platform owner, AI Use Case sponsor, technology lead.
Participants	Business analysts, workflow designers, builders, data stewards, risk, privacy, security, legal, managers, users, and People partners.

Domain Boundary Notes

- Automation owns the technology-enabled execution of repeatable work.
- Factory owns the broader production discipline for AI Use Cases and measured value creation.
- People owns workforce change, reskilling, adoption, and role redesign.
- Leadership owns governance design, risk appetite, and investment decisions.

SOW Test

Pass: A TIO user/client could hire support to identify automation opportunities, design controls, build or coordinate pilots, measure results, and make scale or stop decisions. The work is practical and deliverable.

Top-Down Validation

Executives can see whether automation is creating value, whether risks are controlled, and whether investment decisions are based on evidence instead of enthusiasm.

Bottom-Up Validation

Employees can see which tasks changed, what they still own, how exceptions work, where to get support, and whether automation is helping rather than confusing their work.

08 Resilience

Short Definition: Resilience is the organization's ability to keep technology services secure, reliable, recoverable, compliant, monitored, and trusted as intelligent work becomes more dependent on digital systems.

Plain-English Test: Can the organization identify its critical technology services, protect them, monitor them, recover them, explain who owns failures, and maintain trust when systems, vendors, data flows, or automations break?

Description

Resilience protects the technology foundation that intelligent work depends on. It includes cybersecurity, privacy, access control, service reliability, continuity, incident response, vendor dependency, monitoring, recovery, and the ability to keep operating when something goes wrong.

A mature Resilience capability is not only a security policy or disaster recovery document. It is an operating capability that makes critical technology services visible, owned, tested, monitored, recoverable, and trusted. As AI and automation become more embedded in work, resilience becomes a condition for value.

Why It Matters

- Without Resilience, the organization may build faster systems that are fragile, unsafe, opaque, or impossible to recover when they fail.
- Resilience protects trust, operations, customers, employees, data, reputation, and decision quality.
- It helps leaders use technology, AI, and automation with confidence because failure modes, accountabilities, and recovery paths are understood.

Five Behaviors

Behavior	Meaning
Identify	Name critical technology services, data flows, systems, vendors, automations, and dependencies that the organization cannot afford to lose.
Protect	Apply access controls, security controls, privacy controls, backup, vendor controls, and configuration standards.
Monitor	Observe system health, data flow health, automation behavior, security signals, vendor status, and user-reported issues.
Respond	Detect incidents, assign ownership, communicate clearly, contain harm, restore service, and capture lessons.
Recover	Test recovery plans, fallback paths, business continuity routines, and resilience improvements based on real evidence.

What This Capability Owns

- Critical technology service map.
- Technology resilience, continuity, recovery, and incident routines.
- Security, privacy, access, monitoring, and vendor resilience controls in partnership with specialist functions.
- Operational response to technology failures and automation failures.
- Resilience evidence, testing, lessons learned, and improvement backlog.

What This Capability Does Not Own

- Enterprise risk appetite by itself.
- All legal, regulatory, or compliance interpretation.
- Data quality and lineage by itself.
- Business continuity for non-technology operations by itself.
- Every security policy, if security is a separate enterprise function.

What People Actually Do

People make this capability real by turning technology ideas into repeated decisions, working routines, and operating evidence.

Role	What they actually do
Executive Sponsor	Sets the importance of resilience, funds critical controls, and accepts or rejects major residual risk.
CIO, CTO, CISO, or Technology Lead	Owns technology resilience, controls, response, recovery, and improvement routines.
Service Owner	Owns the reliability, recovery, support model, vendor dependencies, and user communication for a critical service.
Risk, Privacy, Legal, and Compliance	Confirm obligations, reporting needs, regulatory exposure, records, and residual-risk treatment.
Business Owners and Users	Identify service criticality, operational impact, workarounds, and recovery expectations.

Simple activity flow

1. Identify critical technology services and intelligent-work dependencies.
2. Assess security, privacy, reliability, vendor, access, continuity, and recovery risks.
3. Define owners, controls, monitoring, fallback, incident, and recovery routines.
4. Test the routines through scenarios, incidents, tabletop exercises, and technical checks.
5. Review lessons learned and improve resilience based on evidence.

Real-World Example

An organization launches an AI assistant for operational staff. It works well until the underlying data connection fails and the assistant starts giving stale answers. Users do not know whether to trust the output, and no one is sure who owns the issue.

Resilience makes the service criticality, monitoring, data freshness checks, owner, fallback process, user communication, and recovery path explicit. In real life, Resilience shows up when people know what failed, what to do, who owns recovery, and when the service can be trusted again.

Practical Outputs

Output	What it contains
Critical Technology Service Map	A practical map of essential systems, platforms, integrations, automations, vendors, data flows, owners, and business impact.
Resilience and Control Assessment	A review of security, privacy, access, reliability, monitoring, vendor dependency, backup, continuity, and recovery gaps.
Incident and Recovery Playbook	A plain-language playbook for detecting, escalating, communicating, responding, recovering, and learning from technology failures or incidents.

Evidence of Maturity

- Critical services and owners are documented.
- Access, security, privacy, backup, monitoring, and recovery controls are defined.
- Incident and recovery playbooks exist and are tested.
- Vendor dependencies and concentration risks are reviewed.
- Technology failures and automation failures are logged and reviewed.
- Resilience improvements are funded and tracked.
- Users know how to report issues and what to do when systems fail.

Useful Measures

Measure	What it tells us
Service reliability	Availability, incident frequency, severity, recovery time, and recurrence of critical service failures.
Control coverage	Percentage of critical services with owner, monitoring, access controls, backup, recovery plan, and tested playbook.
Trust and recovery	Time to detect, time to restore, user communication quality, and lessons learned completed after incidents.

Maturity Model

Level	Maturity Description
0 Absent	Critical technology services, dependencies, controls, recovery paths, and owners are not meaningfully documented.
1 Initial / Ad-Hoc	Some controls and recovery practices exist, but they depend on individuals and are not tested consistently.
2 Repeatable	Basic resilience routines exist for major systems, but coverage, monitoring, vendor review, and recovery testing are uneven.
3 Defined	Critical services, owners, controls, monitoring, incident routines, and recovery playbooks are documented and used.
4 Managed	Resilience is measured, tested, improved, funded, and connected to risk, technology, vendor, and automation decisions.
5 Optimized	Resilience continuously adapts to changing threats, technology dependencies, vendor risks, AI services, and operational needs.

Typical Owners and Participants

Role	Typical participants
Executive sponsor	CEO, COO, CIO, CTO, CISO, risk executive, city manager, deputy city manager, or equivalent.
Accountable owners	CIO, CTO, CISO, service owners, platform owners, business continuity lead, risk lead.
Participants	Technology operations, security, privacy, legal, procurement, vendors, business owners, communications, managers, and users.

Domain Boundary Notes

- Resilience owns the protection, reliability, recovery, and trust of the technology environment.
- Leadership owns risk posture, accountability, and governance design.
- Data owns data trust, lineage, quality, and model observability.
- People owns communication, adoption, and workforce experience during technology change or disruption.

SOW Test

Pass. A client could hire support to map critical services, assess resilience gaps, define controls, create recovery playbooks, and test response routines. The work is practical, observable, and measurable.

Top-Down Validation

Executives can see what technology services are critical, what risks matter, what controls exist, and how the organization will respond when things fail.

Bottom-Up Validation

Users can see that systems are reliable, support is clear, incidents are communicated, fallback paths exist, and technology failures are not treated as mysteries.

09 Technology Domain Fit and Handoffs

The Technology domain should be read as one connected system. The five capabilities are not separate workstreams that can mature in isolation. They depend on each other and they connect to the other TIO domains.

Technology capability	Primary question	Primary handoff
Architecture	What technology foundation are we building toward?	Receives direction from Leadership and gives structure to Platforms, Integration, Automation, and Resilience.
Platforms	What systems and tools do people use to get work done?	Provides the usable systems that Data, Factory, and People depend on.
Integration	How do systems, data, workflows, and services connect?	Moves governed information across work and enables automation and reporting.
Automation	Which repeatable work can technology perform safely and measurably?	Connects Technology to Factory through practical AI Use Cases and evidence-based improvement.
Resilience	Can the organization trust, protect, recover, and sustain its technology services?	Connects Technology to Leadership risk posture, Data trust, and People confidence.

Domain Interaction With Other TIO Domains

Domain	How Technology connects
Leadership	Leadership sets enterprise direction, governance design, risk appetite, and investment posture. Technology translates these choices into architecture, platforms, integration, automation, and resilience.
Data	Data defines trusted data, stewardship, quality, lineage, model evidence, and observability. Technology provides the systems, platforms, pipelines, integrations, and monitoring environment Data needs.
Factory	Factory turns improvement ideas, AI Use Cases, automation opportunities, and workflow redesign into measured value. Technology provides the technical environment and automation controls that make delivery possible.
People	People ensures adoption, role redesign, fluency, change, and workforce trust. Technology must provide usable tools, clear support, safe automation, and resilient services people can rely on.

Working Summary

Technology is the enterprise capability that makes intelligent work technically possible. Architecture shapes the foundation. Platforms provide the working environment. Integration connects the work. Automation improves the work. Resilience protects the work. The Technology domain is mature when the organization can make technology choices deliberately, use its platforms well, move information across boundaries, automate repeatable work responsibly, and protect the systems that now carry more of the organization's value, judgment, and trust.