
# Capability Guide: Factory Delivery

*Move from Idea to Working Change.*

**Document Version 1.0 FINAL | TIO-KR-2026.08.0 | August 2026**

# Table of Contents



### How to use this guide

Read Sections 1 through 3 for the executive view. Use Sections 4 through 7 to build and mature the capability. Use Section 8 to decide what to do next Monday morning.

# 01 Executive Brief

Delivery is the organization's ability to move an approved AI Use Case, automation, or work improvement from design to a stable working release.

Delivery is the third Factory capability. Discovery finds. Design shapes. Delivery launches. Measurement proves. Scaling compounds. Delivery is where the organization stops talking about the improvement and makes it real.

| Capability ID | F3 - Delivery |
|---|---|
| **Domain** | Factory |
| **Plain-English test** | Can the organization move from approved design to a working release with clear ownership, testing, controls, launch readiness, user support, and issue resolution? |
| **Five behaviors** | Plan. Build. Test. Launch. Stabilize. |
| **Three outputs** | Delivery Plan. Launch Readiness Checklist. Stabilization and Handoff Record. |
| **Accountable owner** | Factory Lead, product owner, use case owner, delivery lead, or improvement lead, with Technology, Data, Risk, Governance, and People support. |
| **First move** | Confirm the approved design, owner, scope, dependencies, controls, success measures, launch path, and support plan before build or configuration begins. |

### Core idea

Delivery is the organization's ability to build, configure, test, launch, stabilize, and support approved AI Use Cases, automations, and work improvements. It turns a good design into working change that people can use.

**2026.08 ALIGNMENT | Delivery in the refreshed TIO system**

**Primary outcome contribution:** Automation and Enterprise Autonomy

- Build, test, launch, and stabilize the complete operating change, including normal work, exceptions, support, assurance, and recovery.
- Use simulation, shadow, parallel, limited-live, phased, or single-cutover methods according to the transition route and risk.
- Transfer work and authority only when the required evidence and accountable approvals exist.

**Evidence at managed maturity:** Delivery is defined, owned, applied consistently, measured in operation, reviewed through evidence, and improved without weakening trust, accountability, or Enterprise Autonomy.

**First refresh move:** Define the proof method, live boundary, support, stop conditions, and recovery before launch.

# 02 The Problem This Capability Solves

Many organizations have more ideas than working improvements. They approve use cases, hold workshops, create designs, and run pilots. Then the work stalls because nobody owns the delivery path.

Delivery solves this gap. It creates the disciplined path from approved design to working change. It does not need to be heavy. It does need to be clear.

Go-live is not success. Go-live is the start of proof. Delivery must launch the change in a way that users can adopt, issues can be fixed, controls can operate, and Measurement can prove what changed.

## When Delivery is weak

- Good designs remain documents, prototypes, or disconnected pilots.
- Teams build before dependencies, data, controls, and support needs are clear.
- Technology, Data, Risk, Governance, and People are brought in too late.
- Testing focuses on whether the tool works, not whether the work works.
- Users receive a new tool or workflow without enough training, support, or ownership.
- Issues after launch are treated as surprises instead of expected stabilization work.
- Measurement cannot prove value because launch evidence and operating ownership were weak.

# 03 What this Capability Is

Delivery turns a designed use case into a working release. It coordinates build, configuration, testing, launch, stabilization, and early support so the redesigned work can operate in the real organization.

A mature Delivery capability is disciplined without becoming bureaucratic. It protects value by making sure the use case is built on approved platforms, uses the right data, includes required controls, prepares users, and has an owner after launch.

## What Delivery owns

- Use case delivery planning for the approved design.
- Build or configuration coordination for the use case or improvement.
- Testing of the workflow, outputs, controls, exceptions, user experience, and support path.
- Launch readiness and release coordination.
- Early stabilization, issue resolution, and handoff to Measurement and ongoing owners.

## What Delivery does not own

- Enterprise strategy, portfolio funding, or governance decision rights. Leadership owns those choices.
- Enterprise platform strategy, integration standards, security operations, or system resilience. Technology owns those capabilities.
- Data ownership, quality, access, lineage, or flow rules. Data owns those capabilities.
- Risk appetite, formal risk assessment, and control expectations. Risk owns those inputs.
- Adoption system, role changes, workforce readiness, and culture. People owns those capabilities.
- Value proof and Evidence Gate recommendations after launch. Measurement owns those decisions with the use case owner.

## Five core behaviors

| Behavior | What it means in practice |
|---|---|
| **Plan** | Confirm scope, owner, roles, dependencies, success measures, controls, release path, and support plan. |
| **Build** | Configure, assemble, or coordinate the solution using approved platforms, data, technology patterns, and delivery methods. |
| **Test** | Test the work, not only the tool. Check outputs, controls, exceptions, access, user experience, and support readiness. |
| **Launch** | Release the improvement with communication, training, ownership, issue management, and clear go-live criteria. |
| **Stabilize** | Fix early defects, monitor use, support users, resolve issues, and prepare a clean handoff to Measurement and ongoing owners. |

# 04 How the Capability Works

Delivery starts after Design has created an approved, buildable future-state design. It ends when the release is stable enough to measure and operate.

## Simple operating model

| Step | Plain-English question |
|---|---|
| **1. Confirm readiness** | Is the design approved, owned, funded, scoped, and ready to build? |
| **2. Prepare delivery plan** | Who will do what, by when, with which dependencies, controls, and success measures? |
| **3. Build or configure** | Can the solution be built using approved platforms, data, access, integration, and automation patterns? |
| **4. Test the whole workflow** | Does the improved work perform safely and reliably for real users and real exceptions? |
| **5. Prepare launch** | Are users, owners, support teams, communications, controls, and issue paths ready? |
| **6. Launch and stabilize** | What defects, adoption issues, data issues, or control issues must be fixed before Measurement begins? |
| **7. Handoff** | Who owns the use case after launch, what evidence is available, and what will Measurement test? |

## The Delivery rule

**Do not confuse launch with value.**

A release can go live and still fail to create value. Delivery is successful when the use case is stable enough for users to operate it and for Measurement to prove what changed.

## How Delivery connects to the rest of TIO

- Leadership provides priority, sponsorship, decision rights, and risk posture.
- Data provides ownership, quality, access, flow, and trusted data foundations.
- Technology provides approved platforms, integration, automation patterns, environments, security, and resilience.
- Factory provides the path from Discovery and Design into Delivery, Measurement, and Scaling.
- People provides communication, role clarity, training, readiness, adoption, and workforce support.

# 05 Outputs, Evidence, and Measures

Delivery should produce a small number of useful artifacts that make the release clear, testable, supported, and ready for Measurement.

| Output | Purpose | Owner | Quality standard |
|---|---|---|---|
| **Delivery Plan** | Defines scope, roles, timeline, dependencies, controls, testing, launch path, and support responsibilities. | Delivery Lead and Use Case Owner | Short, current, realistic, and connected to the approved design. |
| **Launch Readiness Checklist** | Confirms that testing, access, controls, training, support, communications, and issue paths are ready. | Delivery Lead with Technology, Data, Risk, Governance, and People input | Used before release. Not a formality. |
| **Stabilization and Handoff Record** | Captures defects, issues, fixes, support needs, owner handoff, and evidence for Measurement. | Use Case Owner and Factory Lead | Shows whether the release is stable enough to measure and operate. |

## Evidence of maturity

- Approved Delivery Plan tied to the Design Brief.
- Named use case owner, delivery lead, business owner, and support owner.
- Dependency, risk, control, data, access, integration, and support checks.
- Workflow, output, exception, control, and user acceptance testing records.
- Launch communication, training, and support materials.
- Issue log, defect log, stabilization notes, and handoff record.
- Measurement-ready baseline, success measures, and early operating evidence.

## Useful measures

| Measure | What it tells leaders |
|---|---|
| **Approved designs moved into Delivery** | Whether the Factory is moving from design to action. |
| **Delivery cycle time** | Whether useful work is moving at a practical pace. |
| **Launch readiness defects** | Whether readiness problems are caught before release. |
| **Post-launch issue volume and severity** | Whether the release is stable and supportable. |
| **User adoption in the first 30 days** | Whether people are actually using the improvement. |
| **Handoff completeness** | Whether ongoing ownership and Measurement are ready. |
| **Rework caused by unclear design or missed dependencies** | Whether Discovery, Design, Technology, Data, Risk, Governance, or People handoffs need improvement. |

# 06 Maturity: 0 to 5

| Level | What it looks like | Evidence | Next move |
|---|---|---|---|
| **0 - Absent** | Ideas and designs do not have a clear path to working release. | No delivery plan, launch checklist, or named delivery owner. | Pick one approved design and create a simple delivery path. |
| **1 - Initial** | Delivery happens through individual effort and informal coordination. | Ad hoc builds, uneven testing, unclear support, and weak handoff. | Name delivery roles and use a basic launch readiness checklist. |
| **2 - Repeatable** | A basic delivery pattern exists for selected use cases. | Delivery plan, testing notes, launch checklist, issue log, and owner handoff exist for priority use cases. | Apply the pattern consistently and connect it to Design and Measurement. |
| **3 - Defined** | Delivery is documented, owned, communicated, and used consistently. | Clear roles, release gates, support paths, testing standards, and stabilization routines. | Measure cycle time, defects, readiness, adoption, and rework. |
| **4 - Managed** | Delivery is measured, improved, and connected to portfolio decisions. | Dashboards or reviews show delivery flow, dependencies, issues, readiness, adoption, and handoff quality. | Use delivery evidence to improve Factory throughput and cross-domain handoffs. |
| **5 - Optimized** | Delivery continuously improves and helps the organization launch safer, faster, and more reliable change. | Patterns are reused, weak launches decline, users are supported, and delivery learning improves future use cases. | Standardize reusable delivery playbooks and scale proven patterns. |

# 07 How to Build and Mature the Capability

Start small. Delivery does not need a large program office. It needs clear ownership, a practical plan, real testing, launch readiness, stabilization discipline, and a clean handoff to Measurement.

## Minimum viable Delivery capability

- One named owner for each use case entering Delivery.
- A one-page Delivery Plan tied to the approved Design Brief.
- A simple launch readiness checklist.
- A test plan that checks the workflow, outputs, access, controls, exceptions, and support path.
- A visible issue log during build, launch, and stabilization.
- A handoff record that prepares the use case for Measurement.

## First 30 days

- Choose one approved Factory Design to deliver.
- Confirm scope, owner, delivery lead, dependencies, controls, and success measures.
- Create the Delivery Plan and Launch Readiness Checklist.
- Identify Technology, Data, Risk, Governance, and People inputs needed before launch.
- Define what must be true before the use case is considered ready to release.

## First 90 days

- Deliver one or two priority use cases through the same delivery pattern.
- Track defects, launch issues, adoption issues, support needs, and rework.
- Run a short post-launch stabilization review.
- Refine the Delivery Plan, checklist, and handoff record based on what was learned.
- Connect Delivery evidence to Measurement and the next Evidence Gate.

## First year

- Create a reusable Delivery playbook for common use case types.
- Build a repeatable launch readiness and stabilization rhythm.
- Track delivery flow, readiness issues, post-launch defects, and adoption signals.
- Strengthen handoffs from Design and into Measurement.
- Decide which delivery patterns can be reused, improved, or retired.

# 08 TSLA Support and First Moves

TSLA helps clients turn approved use cases and designs into working change without overbuilding the delivery model. The goal is practical delivery discipline that protects value, trust, users, and capacity.

## Where TSLA can help

- AI Use Case Delivery Plan design.
- Launch Readiness Checklist and stabilization routine.
- Use Case Owner and Delivery Lead coaching.
- Cross-domain delivery review with Technology, Data, Risk, Governance, and People.
- Pilot stabilization review and handoff to Measurement.
- Thought-Partnership support for early Factory delivery.

## What to do Monday morning

- Pick one approved Design that is ready to move into Delivery.
- Name the use case owner, delivery lead, support owner, and business sponsor.
- Create a one-page Delivery Plan.
- Create a Launch Readiness Checklist before build or configuration begins.
- Ask Technology, Data, Risk, Governance, and People what must be true before launch.
- Start an issue log immediately.
- Define what evidence Measurement will need after stabilization.

**Final thought**

Delivery is where the Factory becomes real. The work is not successful because it was launched. It is successful when people can use it, issues are handled, ownership is clear, and evidence can show what changed.

# A Source Note

This guide is part of the TIO Library. It is aligned to the TIO Framework, the TIO Factory Capability Domain Master, the TIO Capability Model and Maturity Assessment Tool, and the approved 8-section Capability Guide structure used across the TIO Library. It translates the Factory Delivery capability into practical guidance for leaders and practitioners.

It is written as a practical executive guide. It is not a technical implementation standard, legal opinion, privacy assessment, cybersecurity review, or professional assurance report. Organizations should apply their own policies, obligations, accountabilities, and professional review where required.