BEST-PRACTICE GUIDE · TRANSFORM

Phase 07 — Deploy

From quality evidence to controlled production operations

The Deploy phase takes the tested solution into production with minimal risk: declarative infrastructure, GitOps-based delivery and progressive rollout strategies (canary, blue-green) with automatic rollbacks. Legacy replacements follow a staged cutover plan with parallel runs.

Goal of the phase

The phase delivers the productive target solution with proven operability: an automated release track, working rollback paths, a completed cutover and a documented system handed over to operations.

Process model

01 **Release & cutover planning**
Define rollout strategy, cutover windows, data migration steps and abort criteria per application; align with business, operations and change management.

02 **Infrastructure & environments (IaC)**
Declarative provisioning of all environments as infrastructure as code; environment parity from test to production, secrets and certificate management.

03 **Progressive delivery**
GitOps-based deployments with canary or blue-green strategy; automated health checks and metric-based promotion or rollback.

04 **Data migration & parallel run**
Staged data migration with reconciliation; parallel run of legacy and new system until equivalence is proven at production scale.

05 **Cutover & handover to operations**
Controlled cutover with a hypercare phase; handover of runbooks, dashboards and escalation paths to the Operate phase.

Methodology mix

METHOD	PURPOSE	REFERENCE
Continuous deployment & deployment pipeline	Fully automated release track with staged quality gates; foundation of short, low-risk release cycles.	Humble & Farley (2010)
GitOps	Declarative desired state in version control with automatic reconciliation; auditable, reproducible deployments.	OpenGitOps / CNCF
Progressive delivery (canary, blue-green)	Stepwise traffic shifting with metric-based promotion; limits the blast radius of failures.	Google SRE
Infrastructure as code	Versioned, testable infrastructure definitions; eliminates configuration drift between environments.	Morris (2020)

METHOD	PURPOSE	REFERENCE
Strangler fig migration	Incremental replacement of legacy systems behind a facade; avoids big-bang cutovers.	Fowler (2004)
DORA change management	Empirically validated practices for low-risk changes; measured via change failure rate and MTTR.	DORA / Forsgren et al.

Platform support

ReqPOOL Suite: **Signoff** (AI-based acceptance) · **REAM** (Enterprise architecture)

Signoff — user acceptance & sign-off	Guided playbooks for business users with multimodal evaluation of the screen recordings; an independent AI verification model assesses every test case — the final acceptance stays with the client.
Signoff — acceptance report as go/no-go basis	The revision-proof acceptance report with full traceability from specification clause to test result is the documented basis of the go-live decision — robust towards suppliers and auditors alike.
REAM — cut-over steering	Cut-over plan, data migration and interface switching; parallel operation is reconciled against the AS-IS baseline.
REAM — controlled legacy decommissioning	Decommissioning of legacy systems on the target date; the landscape inventory is updated with the go-live.

Best-practice checklist

- [ ] Start the cutover only once the Signoff acceptance report is available without open critical findings (gate: acceptance granted).
- [ ] Define and technically rehearse abort criteria and rollback paths before every rollout — an untested rollback is not a rollback.
- [ ] Reconcile the parallel run against the AS-IS baseline from the Analyse phase; unresolved deviations block the legacy decommissioning.
- [ ] Document go/no-go decisions exclusively in the cutover protocol — verbal approvals are inadmissible.
- [ ] Agree the hypercare period and staffing before the cutover; hand over to regular operations only with accepted runbooks.
- [ ] After go-live, update the target landscape in REAM and follow up the legacy decommissioning to the target date.

Artifacts & outcomes

— Automated release track with progressive rollouts
— Infrastructure as code for all environments
— Completed data migration with reconciliation evidence
— Cutover protocol with go/no-go decisions
— Operations handover package (runbooks, dashboards, escalation paths)

Quality gate criteria

☐	Production stable over the agreed hypercare period
☐	Rollback capability proven
☐	Data reconciliation without open critical deviations
☐	Operations handover confirmed by the Operate team

Scientific deep dives

Humble & Farley: Continuous Delivery (2010) — Standard reference for deployment pipelines and release management.
https://continuousdelivery.com/

Google SRE Workbook: Canarying Releases — Practice guide for metric-based progressive rollouts.
https://sre.google/workbook/canarying-releases/

OpenGitOps (CNCF) — Principles and standards of the GitOps approach.
https://opengitops.dev/

Fowler: Strangler Fig Application — Pattern of incremental legacy replacement.
https://martinfowler.com/bliki/StranglerFigApplication.html

Kim, Humble, Debois & Willis: The DevOps Handbook — Comprehensive reference on DevOps principles and flow optimization.
https://itrevolution.com/product/the-devops-handbook-second-edition/

DORA: capability catalogue — Research-based capabilities for delivery performance.
https://dora.dev/capabilities/