Faysal Aziz

Process Automation

for GBS Teams with No-Code AI

FA-0365

Agentic & Generative AI · Automation, RPA & Power Platform

Introduction

Process automation in shared services and GBS has shifted from “big IT programs” to a practical operating capability that business teams can build and run themselves. In 2026, the fastest path to measurable cycle-time reduction isn’t a monolithic platform roll-out—it’s assembling a small, repeatable set of automation patterns (intake → triage → decisioning → execution → audit) and deploying them with no-code AI tools that can read, classify, route, and act across the apps you already use.

At the same time, the stakes are higher: modern “agentic” tools can operate email, browsers, and enterprise systems with real permissions, which makes governance, access control, and auditability part of the automation design—not an afterthought.

This course is designed for non-technical operations professionals: team leads, process owners, QA, workforce/MI, and transformation roles. The instructor has over 30 years of industry experience and teaches what’s demanded in real operations environments—measurable outcomes, controls, and adoption—rather than academic theory.

Learning outcomes

By the end of the course, learners will be able to:

- Identify, select, and prioritize automation candidates in P2P, O2C, R2R, HR ops, and service desk processes.
- Map processes at the “automation-ready” level: triggers, inputs, decisions, exceptions, SLAs, and handoffs.
- Build no-code automations using AI steps for classification, extraction, summarization, and decision support.
- Design human-in-the-loop controls (approvals, confidence thresholds, exception queues) that reduce risk while keeping speed.

- Implement governance for citizen development: access, secrets, data handling, change control, and audit evidence.
- Evaluate and safely deploy agent-style tools (including self-hosted options) with permission boundaries and monitoring.
- Create a reusable automation playbook and KPI model (cycle time, touch time, rework, accuracy, cost-to-serve, compliance).

Prerequisites

- Comfort with basic process concepts (SIPOC or similar) and operational metrics.
- Familiarity with common business tools (email, spreadsheets, ticketing, shared drives).
- No coding required.
- Helpful but not required: experience in shared services/GBS operations or continuous improvement.

Training outline

1. Process automation in GBS: what changes with no-code AI

1. From task automation to end-to-end “workflow automation”
2. Where AI fits: understanding, deciding, and generating vs. deterministic execution
3. The “agent” shift: tools that can operate software like a user
4. Value model for shared services: cost-to-serve, speed, quality, compliance, experience
5. Operating model choices: centralized CoE, federated citizen dev, hybrid

2. Automation discovery for non-tech teams

1. Building an automation inventory across towers (P2P, O2C, R2R, HR, master data, service desk)
2. Selecting candidates: volume, variance, system landscape, risk profile, exception rate
3. Defining “done”: measurable outcomes and baseline metrics
4. Process maturity checks: standard work, inputs quality, ownership, policy clarity
5. Intake and demand shaping: avoiding “random acts of automation”

3. Automation-ready process design

1. Event triggers: time-based, message-based, record-based, human request
2. Data inputs: emails, PDFs, spreadsheets, forms, tickets, chat messages

3. Decision points: rules vs. AI classification vs. approval gates
4. Exception handling: queues, reason codes, reroute paths, escalation rules
5. Evidence and audit trail requirements (what to capture, where to store it)

4. No-code AI building blocks (tool-agnostic)

1. Connectors and integrations: APIs, webhooks, email parsing, file watchers
2. Structured vs. unstructured data handling

   AI steps

   1. Document understanding: extraction, normalization, validation
   2. Classification and routing: intent, category, priority, owner
   3. Summarization and drafting: notes, customer updates, case wrap-ups
   4. Entity detection: vendor names, invoice IDs, PO numbers, bank details
   5. Confidence scoring and thresholds

   Deterministic steps

   1. Lookups, transformations, deduplication, formatting
   2. Business rules, SLAs, timers, retries, idempotency

   Human-in-the-loop patterns

   1. Approval tasks, exception queues, “review before send”
   2. Segregation of duties for sensitive actions
   3. Sampling strategies and QA controls

5. Core platform patterns for business users

1. Email-to-case automation: triage, classify, route, auto-responses
2. Ticket enrichment: pulling context from files, ERP/CRM, knowledge bases
3. Document-to-transaction: invoice intake, claim intake, KYC-like intake (non-regulated overview)
4. Reconciliation support: variance detection, exception packages, follow-up workflows
5. Master data requests: validation gates, evidence attachment, maker-checker
6. Service request automation: forms, routing logic, SLA tracking

6. Agentic automation (and how to keep it safe)

1. What agent tools do differently: planning, tool use, multi-step execution
2. Permissioning models: least privilege, scoped tokens, role-based access
3. UI automation vs. API automation: when “computer use” is justified
4. Prompting for operations outcomes: constraints, checklists, stop conditions

5. Monitoring and rollback: what to do when an agent goes off-script
6. Security reality check: phishing/social engineering risks, token misuse, and guardrails

7. Tool landscape for this course (no-code and non-technical friendly)

Workflow automation hubs

1. **Zapier**: agent-driven automation across large app ecosystems
2. Visual workflow builders (when to use them for ops orchestration)
3. Self-host and governance-friendly alternatives for enterprises

Microsoft-centric enterprise automation

1. **Microsoft Copilot Studio**: building agents with connectors and multi-channel deployment
2. **Power Automate**: AI-first workflow capabilities and enterprise observability
3. "Computer use" style automation: where it helps, where it creates risk

Emerging agent/action engines

1. **Manus**: executing tasks beyond chat, with multi-tool skills and workflow execution

Clawbot / self-hosted agent tools

1. **Clawbot**: what it is, where it fits (personal/ops assistant patterns), and operational constraints
2. "Look-alike" tool risk and supply-chain hygiene (avoiding fake extensions and unofficial builds)

Complementary platforms and ecosystems

1. Automation + data tables + forms as a lightweight "ops layer"
2. Knowledge bases and SOP systems as automation context

8. Governance for citizen automation in GBS

1. Policy framework: which processes are allowed, which are restricted, which require approvals

   Security controls

   1. Secrets management and connector approvals
   2. OAuth consent policies, admin approval flows, MFA/conditional access
   3. Data loss prevention (DLP) and retention

   Development lifecycle without code

   1. Naming standards, documentation, versioning, change control
   2. Testing approach: happy path, edge cases, exception simulation
   3. Release gates: peer review, maker-checker, sign-offs

   Monitoring and auditability

   1. Logs, evidence capture, decision rationale, exception reasons

2. KPI dashboards: savings, SLA attainment, quality, automation health

Operating model

1. CoE templates, reusable components, approved connector catalog
2. Training and certification path for citizen developers
3. Support model: L1/L2 for automations, incident response playbooks

9. Process measurement and ROI (what GBS leaders actually ask for)

1. Baselines: volume, AHT/touch time, rework, defect rates, backlog, escalations
2. Benefits tracking: hard savings vs. capacity release vs. risk reduction
3. Automation scorecards: reliability, adoption, exception rate, control effectiveness
4. Continuous improvement loop: tune rules, retrain prompts, tighten thresholds

10. Implementation playbook (repeatable delivery)

1. Use-case blueprint: one-page “automation canvas” for non-tech teams
2. Build standards: prompts, rules, exception codes, approval matrices
3. Launch plan: pilot scope, comms, training, hypercare, stabilization
4. Scale plan: factory approach, reuse library, prioritization cadence
5. Ethical and compliance considerations: transparency, accountability, data minimization

---

© 2026 Faysal Aziz · Kuala Lumpur