Quicksort

GENERATED FROM THE ASSESSMENT



BUILD PRD · SOFTWARE FACTORY

Example bank's software factory: from today's levels to the targets

Releases of the customer onboarding service take about two weeks, mostly waiting for checks and approvals. This document sets out what to build, in what order, what stays with people, and how each step will be proven. It is written from the software delivery assessment.

COMPANY	SCOPE	DEPLOYMENT
Example bank, banking	8 delivery steps, 11 jumps in 3 waves	Option C
ANSWERS	**CRITICAL FUNCTIONS**	**STATUS**
50 of 50 questions	In scope	Version 1, for approval

Contents

01 Executive summary
02 The need and the scope
03 Where you are today
04 Where you want to go
05 Rules and constraints
06 People and responsibilities
07 The factory: components
08 Requirements, step by step
09 Cross-cutting requirements
10 Verification and acceptance
11 Rollout plan
12 Governance and change
13 Risks
14 Open questions
15 Sign-off
Appendix A · Answer and evidence register
Appendix B · Factory configuration
Appendix C · The levels

Most steps can move one level. Testing comes first.

STEPS AT L1 OR BELOW	JUMPS PLANNED	DEPLOYMENT
3 of 8	11	Option C
Steps where AI is not yet doing whole tasks	One level at a time, in 3 waves	Sovereign

THE NEED STATED

Releases of the customer onboarding service take about two weeks, mostly waiting for checks and approvals.

Example bank was assessed on the eight steps of software delivery and seven enablers. The weakest steps today are Specification and planning (L1) and Design (L1). Each step's level is set by its weakest practice shared by most teams.

The plan moves 8 steps towards their targets in 3 waves. Some targets need other steps first: Implementation at L4 needs Specification and planning at L3 and Testing and checks at L4 first. Anything above L3 also needs the verification base (E4) largely in place.

People keep every approval that the charter reserves for them. No agent approves its own work or changes to the factory. Review and release stay at L3 for critical functions unless internal audit signs off.

What this PRD asks you to approve

- The targets per step and the order of the waves
- The requirements for wave 1, and the owners named for each step
- The cross-cutting requirements: data, models, agent identity, evidence and independent approval
- The open questions to settle before wave 1 starts

The need and the scope.

The need	Releases of the customer onboarding service take about two weeks, mostly waiting for checks and approvals.
Company	Example bank, banking
In scope	The eight steps of software delivery, from intake to operations, for the teams assessed
First pilot	One team and one service or repository per step, chosen with each owner at kickoff
Deployment	Option C, sovereign: European or open-weight models on your servers or a SecNumCloud-qualified cloud, with no fallback to other models
Critical functions	In scope. Review and release capped at L3 for these systems.

Out of scope

- Building or training AI models
- Replacing the tools teams already use
- Agents approving their own work, or deploying to production without the approvals the charter requires
- Any measurement or scoring of individual employees

Where you are today.

Level today · Teams ahead of the rest · Target · Above the charter ceiling

STEP	L0	L1	L2	L3	L4	L5
P1 Intake and prioritisation	■	■	■	▭	▭	▭
P2 Specification and planning	■	■	▭	▭	▭	▭
P3 Design	■	■	▭	▭	▭	▭
P4 Implementation	■	■	■	┄	▭	▭
P5 Testing and checks	■	■	■	▭	▭	▭
P6 Review and acceptance	■	■	■	▭	▨	▨
P7 Release and change control	■	■	■	▭	▨	▨
P8 Operations and maintenance	■	■	▭	▭	▭	▭

Findings by step

STEP	LEVEL	WHAT IT LOOKS LIKE TODAY	NOTES
Intake and prioritisation	L2	An agent drafts the ticket from a raw request (title, description, type, suggested priority); a person checks each one.	—
Specification and planning	L1	People use approved AI assistants to draft requirements or task lists.	—
Design	L1	People use approved AI assistants for ideas, diagrams or drafts.	—
Implementation	L2	Agents write whole changes (code, tests, documentation) from a task; a developer reviews and fixes each one.	some teams at L3
Testing and checks	L2	An agent writes tests and runs checks for each change; a person checks the tests and results every time.	—
Review and acceptance	L2	A review agent reviews each change; a person reviews every change and makes the decision.	—
Release and change control	L2	An agent prepares the release (notes, checklist, risk summary, rollback plan); a person runs and approves the release.	—
Operations and maintenance	L1	People use approved AI assistants to read logs or explain errors.	—

Enablers

ENABLER	STATUS
E1.1 Tool set and internal platform	Largely in place

ENABLER	STATUS
E2.1 Models and deployment	Fully in place
E3.1 Context and knowledge	Partially in place
E4.1 Verification base	Partially in place
E4.2 Verification base	Partially in place
E5.1 Governance, risk and compliance	Largely in place
E5.2 Governance, risk and compliance	Partially in place
E6.1 People, operating model and measurement	Partially in place
E6.2 People, operating model and measurement	Partially in place
E7.1 Agent security and identity	Partially in place

Tools in use

STEP	TOOLS
Intake and prioritisation	Jira or Jira Service Management, Azure DevOps Boards
Specification and planning	Confluence, Jira or Azure DevOps (epics, stories, tasks)
Design	Figma, Miro, Lucidchart or draw.io
Implementation	IDEs (Visual Studio Code, IntelliJ IDEA, Visual Studio, Eclipse), GitHub Copilot
Testing and checks	Unit test frameworks (JUnit, pytest, Jest, NUnit or similar), End-to-end tools (Playwright, Cypress, Selenium)
Review and acceptance	GitHub pull requests, GitLab merge requests
Release and change control	GitHub Actions, GitLab CI/CD
Operations and maintenance	Datadog, Dynatrace

Where you want to go.

STEP	TODAY	TARGET	MEASURE OF SUCCESS	OWNER
Intake and prioritisation	L2	L3	Median time from a request arriving to it being prioritised and routed	Product or delivery lead
Specification and planning	L1	L3	Share of specifications sent back for rework, and time from request to approved specification	Product or delivery lead
Design	L1	L2	Median time from approved specification to approved design	Engineering lead or architect
Implementation	L2	L4	Median time from task start to merge request opened	Engineering lead or architect
Testing and checks	L2	L4	Median time from merge request to all checks passed	Platform, QA or SRE lead
Review and acceptance	L2	L3	Median review wait time, and share of changes reopened after acceptance	Engineering lead or architect
Release and change control	L2	L3	Median time from approval to production, and change failure rate	Platform, QA or SRE lead
Operations and maintenance	L1	L2	Median time to restore service, and share of alerts handled without a person	Platform, QA or SRE lead

Baselines for every measure are taken from your own tools in the first week of each wave (TO BE MEASURED). No target is set as a number until the baseline exists.

Goals

- Each step reaches its target level, confirmed in the next assessment
- Engineer involvement per shipped change falls wherever the charter allows
- Change failure rate does not get worse as autonomy rises
- Every change has an evidence record an auditor can follow

Rules and constraints.

Dependency rules applied

- No step goes above L3 unless testing and checks are at least at that level and the verification base is largely in place
- Review cannot run ahead of testing; release can be at most one level above review
- Implementation above L2 needs intake and specification at L2; implementation at L4 needs specification at L3
- Operations at L3 needs intake at L3, so incidents become routed work
- Self-improvement (L5) only where a step is stable at L4 with a working replay set

TARGETS ADJUSTED BY THE RULES

Implementation at L4 needs Specification and planning at L3 and Testing and checks at L4 first. Anything above L3 also needs the verification base (E4) largely in place.

Regulation

DORA (EU 2022/2554)	Every change recorded, tested, approved independently of the requester and the implementer, implemented and verified (Art. 9(4)(e); RTS 2024/1774 Art. 17). Changes to the factory itself follow the same cycle.
EU AI Act	The factory is not a high-risk system in itself. It must not allocate work to people or evaluate them (Annex III, point 4). Changes to high-risk systems such as credit scoring are reviewed with compliance.
GDPR	No personal data in prompts, logs or test data unless approved; synthetic data for previews.
Works council	Partially in place. Consultation before any pilot where French or German teams are affected.

Deployment and models

Option C, sovereign: European or open-weight models on your servers or a SecNumCloud-qualified cloud, with no fallback to other models. Each approved model is measured on your own past work before it is used for a step. The factory is not tied to any model provider.

People and responsibilities.

Each step has a named owner. Approvals stay with people; agents prepare, carry out and record the work.

STEP	OWNER	APPROVES	CONSULTED
Intake and prioritisation	Product or delivery lead	Product or delivery lead	Engineering lead
Specification and planning	Product or delivery lead	Product or delivery lead	Engineering lead
Design	Engineering lead or architect	Engineering lead or architect	Engineering lead
Implementation	Engineering lead or architect	Engineering lead or architect	Engineering lead
Testing and checks	Platform, QA or SRE lead	Platform, QA or SRE lead	Engineering lead
Review and acceptance	Engineering lead or architect	Engineering lead or architect, independent of the author	Risk and compliance owner
Release and change control	Platform, QA or SRE lead	Platform, QA or SRE lead, independent of the author	Risk and compliance owner
Operations and maintenance	Platform, QA or SRE lead	Platform, QA or SRE lead	Engineering lead

Sponsor	Owns the targets and the budget; approves this PRD
Risk and compliance owner	Owns the charter; approves every change to the gates and to the factory itself
Quicksort	Builds and runs the factory with your teams until they run it alone

The factory: components.

The factory is assembled from the tools you already run. Each component is configured for the deployment option and the charter.

COMPONENT	WHAT IT DOES	BUILT ON
Intake connector	Brings requests, tickets and alerts into one queue for agents	Jira or Jira Service Management, Azure DevOps Boards
Specification workspace	Where agents draft specifications and acceptance criteria for approval	Confluence, Jira or Azure DevOps (epics, stories, tasks)
Agent runtime	Runs coding and review agents with shared instructions, skills and approved models	IDEs (Visual Studio Code, IntelliJ IDEA, Visual Studio, Eclipse), GitHub Copilot
Sandboxed execution	Isolated environments per run, with outbound traffic restricted	Added by the factory
Model gateway	Routes each step to an approved model, by policy first, then quality, then cost	On-premise or SecNumCloud models
Verification gates	Fail-to-pass checks, static and security analysis, end-to-end tests on previews	Unit test frameworks (JUnit, pytest, Jest, NUnit or similar), End-to-end tools (Playwright, Cypress, Selenium)
Review and approval workflow	Independent review agent and named human approvals at the defined points	GitHub pull requests, GitLab merge requests
Release pipeline	Change approval, deployment, rollback and post-change verification	GitHub Actions, GitLab CI/CD
Monitoring link	Turns incidents, alerts and vulnerabilities into linked work	Datadog, Dynatrace
Evidence log	Append-only record of every item: request, work, checks, model, agent identity, approver	Added by the factory
Replay set and baseline	Past work turned into tasks, with a hidden slice and a frozen baseline, to test every factory change	Added by the factory

Requirements, step by step.

One block per jump in the plan. Requirement IDs read step, target level and number.

WAVE 1

P1 Intake and prioritisation · L2 → L3 (orchestrated)

Today. An agent drafts the ticket from a raw request (title, description, type, suggested priority); a person checks each one.

After this step. Requests from all channels land in one tracker; an agent classifies, deduplicates, links related work and proposes priority; a person confirms priority at a set point.

ID	REQUIREMENT
P1-L3-01	Connect this step to the pipeline, using Jira or Jira Service Management, Azure DevOps Boards
P1-L3-02	Add automated gates that block unproven work
P1-L3-03	Record evidence for every item in one append-only log
P1-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Median time from a request arriving to it being prioritised and routed, against the baseline taken at the start of the wave
Owner	Product or delivery lead

WAVE 1

P2 Specification and planning · L1 → L2 (delegated)

Today. People use approved AI assistants to draft requirements or task lists.

After this step. An agent drafts requirements, acceptance criteria and the task breakdown from the request and existing documentation; an analyst or lead checks each one.

ID	REQUIREMENT
P2-L2-01	Set up shared agent instructions, skills and approved models for this step
P2-L2-02	Run agents in sandboxed environments with short-lived credentials
P2-L2-03	Define how a task is handed to an agent and how its output is accepted

ID	REQUIREMENT
P2-L2-04	Measure usage at team level, never per person

Stays with people	Checking and accepting every output the agent produces; Approving any change to agent instructions or permissions
Acceptance	Every agent output in this step is checked and accepted by a named person, and the record is kept
Measure	Share of specifications sent back for rework, and time from request to approved specification, against the baseline taken at the start of the wave
Owner	Product or delivery lead

WAVE 1

P3 Design · L1 → L2 (delegated)

Today. People use approved AI assistants for ideas, diagrams or drafts.

After this step. An agent drafts a technical design (components, interfaces, data changes) from the specification and the existing code; an architect checks each one.

ID	REQUIREMENT
P3-L2-01	Set up shared agent instructions, skills and approved models for this step
P3-L2-02	Run agents in sandboxed environments with short-lived credentials
P3-L2-03	Define how a task is handed to an agent and how its output is accepted
P3-L2-04	Measure usage at team level, never per person

Stays with people	Checking and accepting every output the agent produces; Approving any change to agent instructions or permissions
Acceptance	Every agent output in this step is checked and accepted by a named person, and the record is kept
Measure	Median time from approved specification to approved design, against the baseline taken at the start of the wave
Owner	Engineering lead or architect

WAVE 1

P5 Testing and checks · L2 → L3 (orchestrated)

Today. An agent writes tests and runs checks for each change; a person checks the tests and results every time.

After this step. Every change passes automated gates: a fail-to-pass test check, static and security analysis, end-to-end tests on a preview; results are recorded as evidence and anything unproven is blocked.

ID	REQUIREMENT
P5-L3-01	Connect this step to the pipeline, using Unit test frameworks (JUnit, pytest, Jest, NUnit or similar), End-to-end tools (Playwright, Cypress, Selenium)
P5-L3-02	Add automated gates that block unproven work
P5-L3-03	Record evidence for every item in one append-only log
P5-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Median time from merge request to all checks passed, against the baseline taken at the start of the wave
Owner	Platform, QA or SRE lead

WAVE 1

P7 Release and change control · L2 → L3 (orchestrated)

Today. An agent prepares the release (notes, checklist, risk summary, rollback plan); a person runs and approves the release.

After this step. Merge and deployment are automated pipelines; the agent prepares the evidence package for each change; a named person independent of requester and implementer approves; rollback is automated; emergency changes are approved after the fact.

ID	REQUIREMENT
P7-L3-01	Connect this step to the pipeline, using GitHub Actions, GitLab CI/CD
P7-L3-02	Add automated gates that block unproven work
P7-L3-03	Record evidence for every item in one append-only log
P7-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves; Capped at L3 for critical functions unless internal audit signs off
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Median time from approval to production, and change failure rate, against the baseline taken at the start of the wave

Owner	Platform, QA or SRE lead

WAVE 1

P8 Operations and maintenance · L1 → L2 (delegated)

Today. People use approved AI assistants to read logs or explain errors.

After this step. An agent investigates an incident or alert and proposes a cause and a fix; a person checks and acts.

ID	REQUIREMENT
P8-L2-01	Set up shared agent instructions, skills and approved models for this step
P8-L2-02	Run agents in sandboxed environments with short-lived credentials
P8-L2-03	Define how a task is handed to an agent and how its output is accepted
P8-L2-04	Measure usage at team level, never per person

Stays with people	Checking and accepting every output the agent produces; Approving any change to agent instructions or permissions
Acceptance	Every agent output in this step is checked and accepted by a named person, and the record is kept
Measure	Median time to restore service, and share of alerts handled without a person, against the baseline taken at the start of the wave
Owner	Platform, QA or SRE lead

WAVE 2

P2 Specification and planning · L2 → L3 (orchestrated)

Today. An agent drafts requirements, acceptance criteria and the task breakdown from the request and existing documentation; an analyst or lead checks each one.

After this step. Specifications follow a shared template; an agent drafts them, checks them against existing rules and systems, flags gaps and contradictions, produces testable acceptance criteria and a linked task breakdown; a named owner approves the version.

ID	REQUIREMENT
P2-L3-01	Connect this step to the pipeline, using Confluence, Jira or Azure DevOps (epics, stories, tasks)
P2-L3-02	Add automated gates that block unproven work
P2-L3-03	Record evidence for every item in one append-only log
P2-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Share of specifications sent back for rework, and time from request to approved specification, against the baseline taken at the start of the wave
Owner	Product or delivery lead

WAVE 2

P4 Implementation · L2 → L3 (orchestrated)

Today. Agents write whole changes (code, tests, documentation) from a task; a developer reviews and fixes each one.

After this step. Agents pick up approved tasks from the pipeline, work in sandboxed environments with shared instructions and skills per repository, and open changes with evidence attached; developers handle what the agent escalates.

ID	REQUIREMENT
P4-L3-01	Connect this step to the pipeline, using IDEs (Visual Studio Code, IntelliJ IDEA, Visual Studio, Eclipse), GitHub Copilot
P4-L3-02	Add automated gates that block unproven work
P4-L3-03	Record evidence for every item in one append-only log
P4-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Median time from task start to merge request opened, against the baseline taken at the start of the wave
Owner	Engineering lead or architect

WAVE 2

P5 Testing and checks · L3 → L4 (governed autonomy)

Today. Every change passes automated gates: a fail-to-pass test check, static and security analysis, end-to-end tests on a preview; results are recorded as evidence and anything unproven is blocked.

After this step. In an allowed domain, the checks are strong enough for evidence to stand on its own: test adequacy is measured (for example mutation testing), reliability is reported as success on every run, and failures route back to the agent automatically.

ID	REQUIREMENT
P5-L4-01	Name the allowed domain in the charter and issue its autonomy certificate
P5-L4-02	Build a replay set from past work, with a frozen baseline
P5-L4-03	Make the checks strong enough for evidence to accept work in the domain
P5-L4-04	Add automatic step-down to L3, rollback and escalation

Stays with people	A named owner of the allowed domain, with a veto; All work outside the allowed domain; Approving changes to the charter and the autonomy certificate
Acceptance	Inside the allowed domain, items are accepted on evidence with a human veto; any incident triggers step-down to L3; scheduled audit samples pass
Measure	Median time from merge request to all checks passed, against the baseline taken at the start of the wave
Owner	Platform, QA or SRE lead

WAVE 2

P6 Review and acceptance · L2 → L3 (orchestrated)

Today. A review agent reviews each change; a person reviews every change and makes the decision.

After this step. An independent review agent, on a different model family from the author, reviews every change with the evidence from P5; a named person independent of the author accepts changes at defined risk points.

ID	REQUIREMENT
P6-L3-01	Connect this step to the pipeline, using GitHub pull requests, GitLab merge requests
P6-L3-02	Add automated gates that block unproven work
P6-L3-03	Record evidence for every item in one append-only log
P6-L3-04	Move human approval to defined points instead of every item

Stays with people	Approving at the defined points in the pipeline; Approving any change to the gates themselves; Capped at L3 for critical functions unless internal audit signs off
Acceptance	Every item passes the automated gates; anything unproven is blocked; every item has an evidence record
Measure	Median review wait time, and share of changes reopened after acceptance, against the baseline taken at the start of the wave

Owner	Engineering lead or architect

WAVE 3

P4 Implementation · L3 → L4 (governed autonomy)

Today. Agents pick up approved tasks from the pipeline, work in sandboxed environments with shared instructions and skills per repository, and open changes with evidence attached; developers handle what the agent escalates.

After this step. In an allowed domain, agents implement changes end to end, including fixes after failed checks, within a set number of attempts; developers handle escalations and work outside the domain.

ID	REQUIREMENT
P4-L4-01	Name the allowed domain in the charter and issue its autonomy certificate
P4-L4-02	Build a replay set from past work, with a frozen baseline
P4-L4-03	Make the checks strong enough for evidence to accept work in the domain
P4-L4-04	Add automatic step-down to L3, rollback and escalation

Stays with people	A named owner of the allowed domain, with a veto; All work outside the allowed domain; Approving changes to the charter and the autonomy certificate
Acceptance	Inside the allowed domain, items are accepted on evidence with a human veto; any incident triggers step-down to L3; scheduled audit samples pass
Measure	Median time from task start to merge request opened, against the baseline taken at the start of the wave
Owner	Engineering lead or architect

Cross-cutting requirements.

ID	AREA	REQUIREMENT
NFR-01	Data boundary	Code, tickets, logs and prompts stay inside the boundary of option C.
NFR-02	Models	Only approved European or open-weight models, on-premise or on a SecNumCloud cloud. No fallback to any other model.
NFR-03	Agent identity	Each agent run has its own short-lived identity with the least access it needs. Status today: Partially in place.
NFR-04	Execution	Agents run in sandboxed environments with outbound traffic blocked by default.
NFR-05	Supply chain	Skills, MCP servers and plugins are pinned, reviewed and allow-listed. Untrusted input is kept apart from instructions.
NFR-06	Independent approval	No agent approves its own work. Production changes are approved by a named person independent of the requester and the implementer.
NFR-07	Fail closed	Missing or failed evidence blocks the item. Nothing passes by default.
NFR-08	Evidence	Append-only log for every item, exportable to your audit tools, kept for the retention period you set.
NFR-09	People data	Metrics at team level only. No measurement, ranking or allocation of individual employees.
NFR-10	Reversibility	Configuration, evidence and replay sets are exportable, so you can take the factory over or leave.
NFR-11	Personal data	No personal data in prompts or test data unless approved; previews use synthetic data.

Verification and acceptance.

Verification is built first, because it limits how far every other step can go. Status of the verification base today: Partially in place (tests and CI), Partially in place (replay of past work).

- **Replay set.** Past merged changes and tickets turned into tasks, keeping those whose tests fail before the fix and pass after it. A hidden slice is never used for tuning.
- **Frozen baseline.** Every change to the factory is compared with the unchanged setup on the same tasks.
- **Layered checks.** Fail-to-pass, static and security analysis, end-to-end tests on previews, an independent review agent on a different model family, calibrated against past human decisions.
- **Reliability.** Reported as success on every run, not success at least once.
- **Audit samples.** Scheduled samples of accepted items are checked by a person; a failed sample triggers step-down.

Acceptance of each wave

A wave is accepted when every jump in it meets its acceptance criterion, the measures are taken against their baselines, and the levels are confirmed in a facilitated session with evidence.

Rollout plan.

Durations are set with your teams at kickoff TO BE SET. Each wave starts only when the previous one is accepted.

WAVE	JUMPS	ENTRY	EXIT
Wave 1	P1 Intake and prioritisation L2→L3 P2 Specification and planning L1→L2 P3 Design L1→L2 P5 Testing and checks L2→L3 P7 Release and change control L2→L3 P8 Operations and maintenance L1→L2	This PRD approved; open questions settled; baselines taken	Acceptance criteria met; levels confirmed with evidence
Wave 2	P2 Specification and planning L2→L3 P4 Implementation L2→L3 P5 Testing and checks L3→L4 P6 Review and acceptance L2→L3	Wave 1 accepted	Acceptance criteria met; levels confirmed with evidence
Wave 3	P4 Implementation L3→L4	Wave 2 accepted	Acceptance criteria met; levels confirmed with evidence

Governance and change.

Charter	Status: Largely in place. The charter sets risk tiers, allowed domains and who approves what. It is signed by the risk and compliance owner before wave 1.
Allowed domains	Each L4 jump names its allowed domain and issues an autonomy certificate: scope, owner, evidence required, step-down triggers, review date.
Step-down	Any incident, anomaly or failed audit sample takes the domain back to L3 automatically.
Changes to the factory	Recorded, tested on the replay set, approved by a person independent of the change, then rolled out in stages.
Works council	Status: Partially in place. Information pack and consultation before any pilot that affects teams covered by a works council.
Skills and roles	Status: Partially in place. Training per wave for the teams in scope; owners named for every step.
Measurement	Status: Partially in place. Team-level measures, including cost per merged change, reviewed at the end of each wave.

Risks.

RISK AND MITIGATION
Flaky checks block work: quarantined and listed in week 1
Review load moves elsewhere: review wait time measured from day one
Agent output accepted without real checking: sample reviews in the first weeks
Agents reaching more than they need: least-privilege, short-lived credentials
Evidence accepts a bad change: automatic step-down to L3 on any incident
Auditors may not accept evidence alone: agreed with internal audit before go-live
On-premise models may perform below frontier models: each is measured on your past work before go-live
Works council not yet consulted: consultation scheduled before the first pilot
Agents lack context (documentation, decision records): context work planned in wave 1
Weak test base: tests that capture current behaviour are written before automation
Agent identities not yet controlled: least-privilege identities set up before any agent runs

Open questions.

All 50 questions are answered. Levels are confirmed with evidence in the facilitated session.

Sign-off.

Approving this PRD approves the targets, the order of the waves, the requirements for wave 1 and the cross-cutting requirements. Each later wave is confirmed before it starts.

SPONSOR	RISK AND COMPLIANCE OWNER	ENGINEERING LEAD
Name, date	Name, date	Name, date

APPENDIX A

Answer and evidence register.

ID	QUESTION	ANSWER	TEAMS	EVIDENCE
P1.Q1	Who or what turns incoming requests into prioritised, routed work?	L2	Most	Evidenced
P1.Q2	Who accepts the priority and routing of new work, and are they independent of whoever drafted it?	L2	Most	Evidenced
P1.Q3	What is recorded about how each item was sorted and prioritised? Could an auditor reconstruct who did what, with which model and which credentials?	L2	Most	Evidenced
P1.Q4	Which security and compliance controls run on agents' work at intake?	L2	Most	Evidenced
P1.Q5	Can the agent find what it needs to sort and route work?	L2	Most	Evidenced
P2.Q1	Who or what writes the specification and breaks the work into tasks?	L1	Most	Evidenced
P2.Q2	Who approves a specification before work starts, and are they independent of its author?	L1	Most	Evidenced
P2.Q3	What is recorded about how each specification was produced? Could an auditor reconstruct who did what, with which model and which credentials?	L1	Most	Evidenced
P2.Q4	Which security and compliance controls run on agents' work in specification?	L1	Most	Evidenced
P2.Q5	Can the agent find what it needs to write a specification?	L1	Most	Evidenced
P3.Q1	Who or what produces the technical design?	L1	Most	Evidenced
P3.Q2	Who approves a design before implementation, and are they independent of its author?	L1	Most	Evidenced
P3.Q3	What is recorded about how each design was produced? Could an auditor reconstruct who did what, with which model and which credentials?	L1	Most	Evidenced
P3.Q4	Which security and compliance controls run on agents' work in design?	L1	Most	Evidenced
P3.Q5	Can the agent find what it needs to design a change?	L1	Most	Evidenced
P4.Q1	Who or what writes code, tests and documentation?	L3	Some teams	Evidenced
P4.Q2	Who accepts an agent's change before it goes to review, and are they independent of the author?	L2	Most	Evidenced
P4.Q3	What is recorded about each change? Could an auditor reconstruct who did what, with which model and which credentials?	L2	Most	Evidenced
P4.Q4	Which security and compliance controls run on agents' work in implementation?	L2	Most	Evidenced

ID	QUESTION	ANSWER	TEAMS	EVIDENCE
P4.Q5	Can the agent find what it needs to implement a change?	L2	Most	Evidenced
P5.Q1	Who or what writes the tests and runs the checks on each change?	L2	Most	Evidenced
P5.Q2	Who decides that a change has passed its checks, and are they independent of the author?	L2	Most	Evidenced
P5.Q3	What evidence is recorded from testing? Could an auditor reconstruct who did what, with which model and which credentials?	L2	Most	Evidenced
P5.Q4	Which security and compliance controls run on agents' work in testing?	L2	Most	Evidenced
P5.Q5	Can the agent find what it needs to test a change?	L2	Most	Evidenced
P6.Q1	Who or what reviews each change?	L2	Most	Evidenced
P6.Q2	Who accepts a change, and are they independent of its author?	L2	Most	Evidenced
P6.Q3	What is recorded about each review and acceptance? Could an auditor reconstruct who did what, with which model and which credentials?	L2	Most	Evidenced
P6.Q4	Which security and compliance controls run on agents' work in review?	L2	Most	Evidenced
P6.Q5	Can the review agent find what it needs to judge a change?	L2	Most	Evidenced
P7.Q1	Who or what prepares and runs a release?	L2	Most	Evidenced
P7.Q2	Who approves a change for production, and are they independent of the requester and the implementer?	L2	Most	Evidenced
P7.Q3	What is recorded about each release? Could an auditor reconstruct who did what, with which model and which credentials?	L2	Most	Evidenced
P7.Q4	Which security and compliance controls run on agents' work in release?	L2	Most	Evidenced
P7.Q5	Can the agent find what it needs to prepare a release?	L2	Most	Evidenced
P8.Q1	Who or what handles alerts, incidents, fixes and upgrades?	L1	Most	Evidenced
P8.Q2	Who accepts a fix or upgrade before it ships, and are they independent of its author?	L1	Most	Evidenced
P8.Q3	What is recorded about each incident, fix and upgrade? Could an auditor reconstruct who did what, with which model and which credentials?	L1	Most	Evidenced
P8.Q4	Which security and compliance controls run on agents' work in operations?	L1	Most	Evidenced
P8.Q5	Can the agent find what it needs to investigate and fix?	L1	Most	Evidenced
E1.1	Tool set and internal platform: do your delivery tools and AI agents connect into one platform?	Largely	—	—

ID	QUESTION	ANSWER	TEAMS	EVIDENCE
E2.1	Models and deployment: which models are approved, and where may they run?	Fully	—	—
E3.1	Context and knowledge: can agents reach the knowledge they need across delivery?	Partially	—	—
E4.1	Verification base: are tests, CI and version control strong enough to trust automated checks?	Partially	—	—
E4.2	Verification base: can past changes be replayed as tasks, and is improvement governed?	Partially	—	—
E5.1	Governance, risk and compliance: is there a charter that sets risk tiers and allowed domains?	Largely	—	—
E5.2	Governance, risk and compliance: have the applicable rules been mapped and the works council consulted?	Partially	—	—
E6.1	People, operating model and measurement: who owns the AI programme, and do people have the skills and roles for it?	Partially	—	—
E6.2	People, operating model and measurement: what is measured, including cost per merged change?	Partially	—	—
E7.1	Agent security and identity: are agents controlled as identities with least privilege?	Partially	—	—

APPENDIX B

Factory configuration.

The starting configuration generated with this PRD. Each build step updates it.

```yaml
# generated with this build PRD
company: example-bank
deployment: option-c
models:
  allowed: approved-on-prem
  fallback: none
steps:
  P1: { today: L2, target: L3 }
  P2: { today: L1, target: L3 }
  P3: { today: L1, target: L2 }
  P4: { today: L2, target: L4 }
  P5: { today: L2, target: L4 }
  P6: { today: L2, target: L3, ceiling: L3 }
  P7: { today: L2, target: L3, ceiling: L3 }
  P8: { today: L1, target: L2 }
tools:
  - jira
  - azure-devops-boards
  - confluence
  - jira
  - figma
  - miro-lucidchart
  - ides
  - github-copilot
  - unit-test-frameworks
  - end-to-end-tools
  - github-pull-requests
  - gitlab-merge-requests
  - github-actions
  - gitlab-ci-cd
  - datadog
  - dynatrace
gates:                  # fail closed
  - fail_to_pass
  - static_and_security
  - e2e_preview
  - independent_review
evidence:
  log: append-only
  export: audit
agents:
  identity: per-run, short-lived
  network: blocked-by-default
people_data: team-level-only
```

APPENDIX C

The levels.

LEVEL	NAME	WHO DECIDES
L0	**Manual.** No AI in this step	People
L1	**Assisted.** People use approved assistants; no whole task goes to an agent	People
L2	**Delegated.** Agents carry out whole tasks; a person checks every output	A person, every time
L3	**Orchestrated.** Agents work in a connected pipeline with automated checks	People, at set points
L4	**Governed autonomy.** Inside an allowed domain, agents complete the work and evidence accepts each change	A named owner per domain; people handle exceptions
L5	**Self-improving.** The factory proposes improvements to itself; a person approves each one. Not work without people.	A person approves every change to the factory