SHIFT Agile Framework Guide

Scalable Hybrid Iterative Framework for Teams

Version 2.0 — Ricardo J. Minas, 2026

Introduction

Most Agile frameworks were designed for a world that no longer exists: co-located teams, predictable delivery, and a manageable human-to-output ratio. The frameworks that followed tried to scale that original model rather than rethink it.

SHIFT was built from a different starting point. It begins with the conditions modern teams actually face.

The Four Failures SHIFT Addresses

The Coordination Failure. When organisations have more than one team, the frameworks break down. Scrum of Scrums is a band-aid. SAFe adds so much ceremony that teams spend more time coordinating than delivering. Most organisations end up with an informal coordination layer of Slack threads, side conversations, and escalations to management, none of which scale or leave a record.

The Innovation Failure. Delivery pressure always wins over exploration. "We'll do the innovation sprint next cycle" never happens. The people closest to the problems, the delivery team, are also the people best positioned to solve them. But they never get the protected space to try. SHIFT makes exploration structurally mandatory, not aspirational.

The Alignment Failure. Teams set OKRs in January and discover in March that the sprint work had no measurable connection to them. The gap between strategy and execution is real, persistent, and rarely discussed honestly. Most frameworks do not have a mechanism for mid-sprint alignment checking. SHIFT does.

The AI Transition Failure. Teams are adopting AI tools within frameworks designed before AI existed. Story points were never designed for a production function where one well-specified task produces ten times the output of a vague one. Velocity charts are misleading when AI capability improves between sprints. The definition of "done" needs to include explicit review steps that did not exist before AI-generated output became common. SHIFT was built knowing that AI-assisted work is the default, not the exception.

SHIFT stands for **Scalable Hybrid Iterative Framework for Teams.** The word "hybrid" is intentional: SHIFT does not demand you discard Scrum or abandon SAFe. It layers on top, replaces what is broken, and fills what is missing.

Part I: Foundation

Chapter 1: The Four Layers of SHIFT

SHIFT operates across four distinct layers. Each layer has a different cadence, different participants, and a different decision scope. The failure mode of most frameworks is that everything lives at one altitude: execution. Strategy bleeds into standup, governance becomes ceremony, and learning is an afterthought.

SHIFT enforces separation of concerns across these layers while providing explicit integration points between them.

Note that Learning is Layer 1, not Layer 4. This is deliberate. In most frameworks, learning is squeezed at the end of a cycle. In SHIFT, it is foundational. Everything else builds on it.

Layer 1: Learning

Cadence: continuous, with structured sprint-end moments. Purpose: capture, synthesise, and route learning back into the delivery system through three channels: operational learning (what slowed us down), product learning (what users actually do), and capability learning (what the team can now do that it could not before). Key artefacts: AI-Augmented Retrospective output, IIS Learning Cards, pattern library updates, SEMI model recalibration.

Layer 2: Delivery

Cadence: 2-week sprints. Purpose: execute committed work, coordinate within the node, maintain DOI alignment, and produce working software. Every item that enters a sprint must have a passing SEMI score. Blockers are classified within four hours. Throughput is tracked daily.

Layer 3: Governance

Cadence: bi-weekly, aligned to sprint boundaries. Purpose: cross-node risk management, dependency resolution, compliance decisions, resource reallocation. Governance in SHIFT is lean by design. It exists to unblock, not to approve. Sessions run for a maximum of 75 minutes.

Layer 4: Alignment

Cadence: quarterly (12-week cycle) with a mid-cycle calibration at week 6. Purpose: connect team delivery to organisational strategy. Set Cycle Objectives. Review portfolio health. Make go/no-go decisions on capability bets. Produces exactly one output per cycle: the Cycle Objective Set with RAG status and DOI Health Scores.

Layer Integration Handshakes

Handshake	From	To	Frequency	Mechanism
Sprint objective	Alignment	Delivery	Sprint start	DOI map, sprint goal
Blocker escalation	Delivery	Governance	As needed	4-hour SLA protocol
Governance signal	Governance	Alignment	Bi-weekly	LGP decision log
Learning injection	Learning	Delivery	Sprint end	Pattern library, SEMI update

---

Chapter 2: Core Principles

People before the process. Every structural choice in SHIFT is evaluated against whether it helps or burdens the people doing the work. Governance, tooling, and ceremony exist to serve teams.

Hybrid by design. Distributed and co-located contributors are treated as equal participants. Asynchronous-first communication is the default. Synchronous time is reserved for decisions and dialogue that require it.

Iterative and adaptive. Short cycles with structured reflection. SHIFT teams do not commit to large plans. They commit to learning loops that progressively sharpen direction.

Lean governance. Oversight and accountability without bureaucracy. Lean Governance Pods keep decisions moving at the pace the work demands, with clear ownership and minimal coordination overhead.

Innovation is part of delivery. Experimentation is not deferred to a future quarter. Innovation-Integrated Sprints build dedicated capacity for exploratory work inside the delivery rhythm so that learning does not compete with shipping.

Continuous alignment. Strategy does not live in a quarterly deck. Dynamic Objectives Integration keeps team-level work connected to organisational goals throughout every sprint, not just at planning.

Part II: Structure

Chapter 3: Adaptive Collaboration Nodes (ACNs)

An **Adaptive Collaboration Node** is the primary delivery unit in SHIFT. It is not a squad, tribe, or chapter. The word "node" is deliberate: it implies connectivity, not isolation. An ACN has clear internal structure, a defined scope of ownership, and explicit interfaces to other nodes.

An ACN owns a **capability domain**, not a feature list. It is responsible for the full vertical slice of work within that domain: from specification through testing through deployment. It is not a feature team or a component team. It owns outcomes, not outputs.

Formation

An ACN is formed when a capability domain is identified that will span more than two cycles, when an existing node has a sustained throughput bottleneck, or when a new AI-assisted workflow requires dedicated governance.

Formation process:

1. **Domain scoping.** The Governance Steward and relevant Anchors define the capability domain in one sentence. If it takes more than one sentence, the domain is too broad.
2. **Anchor assignment.** An Anchor is identified from within the existing team or hired. The Anchor must have delivery credibility in the domain, not just seniority.
3. **Composition drafting.** The Anchor proposes a composition using the Node Composition Matrix. The LGP ratifies.

4. **Interface definition.** The new node and any adjacent nodes produce a one-page interface document: what they consume from each other, what they produce for each other, and how they escalate cross-node issues.

Node Composition Matrix

Team Size	Anchor	Delivery Lead	Contributors	AI Agent Roles	Notes
3	1	0 (Anchor doubles)	2	0-1	Minimum viable. IIS suspended.
5	1	1	3	1	Standard node. Full SHIFT operation.
7	1	1	4-5	1-2	Preferred. Full IIS + two AI agents.
9	1	1	6-7	2-3	Maximum. Consider splitting.

AI Agent Roles are not headcount. They are defined functional roles assigned to AI tooling (for example: "Spec Reviewer Agent" or "Throughput Analyst Agent"). Each AI agent role has a defined input, defined output, and a human owner responsible for its outputs.

Composition Principles

Every Node Contributor must be able to contribute across at least two capability areas within the node's domain. Specialists are welcome, but they cannot be single points of failure.

Each node maintains an **AI Responsibility Map**: a one-page living document that defines, for each contributor, which tasks are AI-assisted, which are AI-led with human review, and which are human-only.

Dissolution

An ACN is dissolved when its capability domain is complete, when throughput signals indicate the node cannot sustain delivery across two consecutive Red DOI cycles, or when the team drops below three people without approved backfill.

Dissolution requires a 2-sprint wind-down period. No new work is taken on. Documentation, handover, and pattern library updates are completed before the node closes. The dissolving node's Learning Cards remain in the shared pattern library permanently.

ACN Visibility

Each ACN maintains a live **Node Health Card**: throughput trend for the last three sprints, DOI alignment status (RAG), current IIS status, and open cross-node dependencies by count and age. The card is updated by the Delivery Lead at sprint end and at the mid-sprint DOI

check. It is readable in 90 seconds. If someone asks for more detail than the Node Health Card contains, the card is not being maintained properly.

ACN Anti-Patterns

The Siloed Node. An ACN that stops sharing its interface document and begins resolving all cross-node dependencies informally. Fix: mandatory interface document review at cycle boundary.

The Permanent Node. An ACN that has completed its capability domain but continues operating because no one formally dissolved it. Fix: dissolution review triggered by the Governance Steward when a node's backlog contains only maintenance items for two consecutive sprints.

The Hero Node. An ACN carries a disproportionate load because it has one or two extremely high-performing contributors, with knowledge transfer not happening. Fix: forced contribution rotation and AI task offload review.

The Phantom AI Node. An ACN that counts AI agent outputs as contributor throughput without a defined human review step. Fix: AI Responsibility Map audit, mandatory human review step for all AI-led outputs.

Chapter 4: Lean Governance Pods (LGPs)

A **Lean Governance Pod** is SHIFT's answer to the governance tax problem. In most scaled frameworks, governance consumes 20 to 40 percent of senior contributor time with minimal delivery impact. LGPs are designed to reduce governance overhead to under 10 percent of any individual's time while maintaining decision quality, traceability, and cross-team coherence.

An LGP is not a steering committee, a management layer, or an approval board. It is a decision-making forum with a defined scope, a time budget, and an explicit anti-bureaucracy mandate.

Composition

Standard LGP (5 to 50 person organisation): - 1 Governance Steward (facilitates, owns the decision log) - 1 Anchor per active ACN (maximum 4 in a single LGP) - 1 product or strategy representative - Optional: 1 external stakeholder as a guest

Hard rule: LGP never exceeds 7 participants. If more than 7 people are needed to make a decision, the decision scope is too broad. Break it into smaller decisions.

Scaled LGP (50+ person organisations, multiple ACNs): LGPs are scoped by capability cluster, not by org chart. Each cluster LGP has a Governance Steward. Governance Stewards from all cluster LGPs form a Portfolio LGP that meets monthly. The Portfolio LGP is the only place cross-cluster decisions are made.

Cadence

Bi-weekly session: 75 minutes maximum. The agenda is fixed.

Emergency LGP: Can be called by any Anchor or Governance Steward. Maximum 30 minutes. Single agenda item only. Outcome: unblock or escalate.

Async LGP: For clearly bounded, low-risk decisions, the Governance Steward runs a 24-hour async decision window. If any participant flags for sync discussion, the item moves to the next bi-weekly session.

Fixed Bi-Weekly Agenda

Slot	Duration	Purpose
Pre-read acknowledgement	5 min	Confirm all participants read the pre-read
Node Health Review	15 min	RAG status across all nodes, flag Reds
Decisions: Unblock	20 min	Cross-node or external blockers requiring action
Decisions: Approve	15 min	Items requiring LGP ratification
Decisions: Defer	10 min	Items not ready: assign owner and due date
DOI Alignment check	5 min	Confirm sprint objectives still align to cycle goals
Parking lot	5 min	Items not in pre-read: logged only, not discussed today

Non-negotiable rule: If an item is not in the pre-read, it is not discussed today. It enters the parking lot and goes on the next pre-read. This single rule prevents LGP from becoming a fire-fighting forum.

Decision Types

SHIFT defines exactly three decision types for LGP. Any item that does not fit one of these three types does not belong in LGP.

Unblock: A delivery blocker requiring cross-node coordination, external engagement, or resource reallocation. The LGP resolves it or assigns an owner with a deadline.

Approve: A decision that has been prepared, pre-read, and requires formal ratification. Includes: new node formation, dissolution proposals, IIS theme ratification, cycle objective adjustments, external dependency commitments. Approvals must have a proposal document of maximum two pages in the pre-read.

Defer: An item that is not ready for decision. The LGP assigns an owner, a due date, and specifies the exact information missing. Defer is not “we’ll discuss later.” It is “this person will do this thing by this date so we can decide.”

Three-strikes rule: Any item that appears in the pre-read three consecutive times without resolution is automatically escalated to the Alignment layer.

Pre-Read Protocol

The pre-read is the most important governance artefact in SHIFT. A well-run LGP spends zero time getting context and 100 percent of its time making decisions.

Pre-read sections (mandatory): 1. Node Health Cards (one per active ACN, maximum one page each). 2. Decision items: for each item, exactly three fields: context, options (two or three with trade-offs), and recommendation. 3. DOI alignment summary: current RAG status per cycle objective. 4. Open items from last session: status update on previously deferred or assigned items.

Pre-read distribution: 24 hours before the LGP session. No exceptions.

LGP Anti-Patterns

The Approval Machine. LGP spends all its time approving low-stakes items that should be delegated. Fix: define a delegation threshold. Decisions below a defined risk or cost threshold are delegated to the Anchor with retrospective LGP notification.

The Status Report Forum. First 30 minutes of every session used for verbal status updates. Fix: enforce pre-read protocol strictly.

The Invisible LGP. Decisions made informally outside the LGP and ratified without discussion. Fix: the Governance Steward maintains a public decision log visible to all ACNs.

The Capture Problem. LGP is dominated by one Anchor or product stakeholder who pre-shapes all decisions. Fix: require that recommendations in the pre-read be authored by someone other than the primary beneficiary.

Chapter 5: Roles

SHIFT has four core roles. In small teams, these roles collapse. In large teams, they remain distinct. Roles are defined by accountability, not by job title.

SHIFT Anchor

The Anchor is the single accountable person for an ACN's capability domain **outcomes**, not outputs. If the node delivers software that nobody uses, the Anchor is accountable for identifying that signal and changing course.

Core accountabilities: - Defining and communicating the node's capability domain clearly enough that every contributor can make prioritisation decisions independently. - Maintaining the DOI connection between sprint work and Cycle Objectives. - Making final prioritisation decisions within the node without escalation. - Representing the node in LGP and at the Cycle Portfolio Review. - Owning IIS theme selection and ensuring Learning Cards are completed. - Sponsoring the node's AI Responsibility Map.

What the Anchor is not: The Anchor does not write user stories, track tasks, or make technical decisions. Technical decisions are made by the contributors best placed to make them.

Time allocation: An Anchor in a 5 to 7 person node should spend 30 to 40 percent of their time on direct delivery work: writing specs, reviewing outputs, pairing on complex problems. An Anchor who spends 100 percent of their time in meetings is disconnected from delivery reality.

Delivery Lead

The Delivery Lead is the operational heart of an ACN. They are responsible for the throughput, coordination, and health of the node's delivery system, sprint over sprint.

Core accountabilities: - Owning the sprint plan: ensuring all work enters the sprint with a passing SEMI score, capacity is accurately planned, and the sprint goal is clear. - Maintaining the Sprint DOI Map and updating RAG status. - Running the mid-sprint DOI check and the daily AI-first standup. - Escalating blockers within the 4-hour SLA. - Maintaining the Node Health Card. - Facilitating the sprint retrospective with AI tooling support. - Tracking throughput and producing the Monte Carlo forecast at sprint end.

What the Delivery Lead is not: Not a Scrum Master, not a manager, not a blocker absorber. Their job is to surface blockers and route them to resolution, not to personally resolve all of them.

Rotation: Delivery Leads can and should rotate at cycle boundaries. A Delivery Lead who has been in the same node for more than two cycles may become a single point of failure for process knowledge.

Node Contributors

Node Contributors are the practitioners doing the primary delivery work. They are defined by their commitment to the node's capability domain and their ownership of work items end to end.

Core accountabilities: - Writing or contributing to specs that meet the SEMI threshold before work enters the sprint. - Updating throughput signals daily. - Contributing to IIS themes with genuine engagement. - Authoring or co-authoring Learning Cards. - Owning the human review step for any AI-led outputs they are responsible for. - Contributing to and maintaining the node's prompt library.

AI-native norms: Every Node Contributor defines their personal AI workflow and documents it in the AI Responsibility Map. They commit to reviewing all AI-generated outputs before treating them as complete. They flag when AI outputs are consistently incorrect in a particular domain, as this feeds into the Learning layer.

Governance Steward

The Governance Steward is responsible for the health of the governance **system**, not for making governance decisions. This distinction is critical.

Core accountabilities: - Compiling and distributing the LGP pre-read 24 hours before each session. - Facilitating the bi-weekly LGP session. - Maintaining the decision log (public, searchable, permanent). - Tracking deferred items and ensuring they have owners and due dates. - Running the three-strikes escalation. - Maintaining the cross-node dependency map. - Owning the Maturity Model self-assessment process.

What the Governance Steward is not: Not a gatekeeper. Not an admin function. This role requires judgement and authority. A junior person assigned to "take notes" is not a Governance Steward.

Time allocation: In a standard organisation, the Governance Steward role requires 20 to 30 percent of one person's time.

Role Flexibility for Small Teams

Team Size	Role Combinations	Notes
3 people	Anchor + Governance Steward combined; Delivery Lead + primary Contributor combined	IIS suspended. LGP replaced by a weekly 30-min external alignment meeting.
5 people	Anchor dedicated; Delivery Lead dedicated; Steward function shared between them; three Contributors	Full SHIFT operation viable.
7+ people	Full role separation	All four roles are held by distinct people.

Anchor and Delivery Lead should not be combined in teams above five people. The authority concentration creates delivery blind spots.

Part III: Rhythm

Chapter 6: The SHIFT Lifecycle

SHIFT runs in 2-week sprints grouped into **12-week cycles** (six sprints per cycle). The cycle is the primary strategic alignment unit. The sprint is the primary delivery unit.

Sprint Structure (10 Working Days)

Day	Activity
Day 1	Sprint Kickoff, Spec Review Session
Days 1-10	Daily AI-First Standup (15 min, async-first)
Day 6-7	Mid-Sprint DOI Check-in
Day 10	Sprint Review (includes IIS Review if applicable)
Day 10	AI-Augmented Retrospective

Cycle Structure (6 Sprints, 12 Weeks)

Sprint	Type	Notes
Sprint 1	Delivery	Cycle objectives set, DOI map initialised
Sprint 2	Delivery	LGP bi-weekly cadence active
Sprint 3	Delivery + IIS	First IIS of cycle (15% capacity)
Sprint 4	Delivery	Monte Carlo forecast updated
Sprint 5	Delivery	Mid-cycle DOI calibration
Sprint 6	Delivery + IIS + Cycle Review	IIS Review, Cycle Portfolio Review (LGP)

At the close of each cycle, the LGP conducts the Cycle Portfolio Review: objectives are re-validated, ACN composition is reviewed, IIS themes for the next cycle are ratified, and Maturity Model self-assessment is run.

Chapter 7: Innovation-Integrated Sprints (IIS)

Innovation-Integrated Sprints are SHIFT's mechanism for sustaining exploratory work within a delivery-focused framework, without the failure modes of 20 percent time, hackathons, or separate innovation teams.

Why IIS Works Where 20% Time Does Not

Problem	20% Time	IIS
Time protection	Informal, first to be cut under delivery pressure	Formally allocated, requires LGP approval to reduce
Outputs	No promotion pathway, ideas die in isolation	Defined funnel: Learning Card, Pilot, Investment proposal
Collective intelligence	Individual projects	Node-level theme, shared Learning Cards
Theme selection	Personal interest, disconnected from strategy	DOI-connected theme selection at cycle kickoff

Capacity Allocation

Standard: 15% of sprint capacity (approximately 1.5 days per person per 2-week sprint).
Minimum: 10%. Below this, IIS is performative and should be formally suspended.
Maximum: 25% for a designated innovation sprint, requiring LGP approval. Cannot occur in consecutive sprints.

IIS capacity is subtracted from total node capacity before sprint planning. It is not a "leftover" activity. It is planned first.

ACN Size	Sprint Capacity	IIS @ 15%
3 people	30 days	4.5 days
5 people	50 days	7.5 days
7 people	70 days	10.5 days

Theme Selection

IIS themes are not individual projects. They are shared exploratory directions. A valid IIS theme candidate has three elements: a question (what are we trying to learn?), a connection (how does this connect to a Cycle Objective or known team limitation?), and a testability statement (how will we know in 1.5 days if this is worth pursuing?).

Learning Cards

A Learning Card is the primary output of an IIS sprint. It is structured, not freeform.

Learning Card format: - **Theme question explored** and sprint reference. - **What we did:** two to four sentences on the specific experiment conducted. - **What we learned:** the most important finding, stated as a declarative sentence. If the team cannot state one clear learning, the theme was not testable enough. - **What we did not learn:** explicitly capturing open questions. This prevents false confidence. - **Signal:** Positive, Neutral, or Negative. - **Recommendation:** one of four options: Promote to Pilot, Continue Exploring, Archive, or Share Externally. - **SEMI score for next step** (if promoting), to give the promotion decision a cost signal.

The Promotion Funnel

1. **IIS Sprint** — Learning Card is created with a Positive signal.
2. **IIS Review** (sprint end, 20 minutes) — Learning Cards are presented and decisions made.
3. **Decision: Promote to Pilot?** — If not: Archive or Continue Exploring. If yes: proceed.
4. **Pilot Sprint** (1 sprint) — Defined success criteria, standard delivery capacity, not IIS capacity.
5. **Pilot Review** (Anchor + LGP decision) — Success criteria assessed.
6. **Decision: Investment Proposal?** — If not: Archive with full data. If yes: proceed.
7. **LGP Approval** — Item enters next cycle's DOI map as a Cycle Portfolio Investment.

Pilot rules: A pilot uses standard delivery capacity, not IIS capacity. Success criteria are written before the pilot begins. If criteria are not met, the pilot is archived. Extensions require a new LGP approval.

Chapter 8: Dynamic Objectives Integration (DOI)

Dynamic Objectives Integration is SHIFT's mechanism for keeping sprint-level delivery connected to cycle-level strategy in real time. The core problem DOI solves: in most

frameworks, misalignment between strategic plan and delivery reality is only discovered at quarter end. DOI introduces a continuous signal system that surfaces misalignment during the sprint, when something can still be done about it.

DOI is not OKRs. It is the operational layer that connects OKRs (or any strategic objective system) to sprint delivery.

The DOI Map

A DOI Map exists at two levels.

Cycle DOI Map (Alignment layer, quarterly): Each Cycle Objective lists two to five Sprint Contributions, each tagged with a RAG status. A Cycle Objective with more than five Sprint Contributions is too broad and should be split.

Sprint DOI Map (Delivery layer, updated each sprint): A subset of the Cycle DOI Map showing only the current sprint's contributions with daily-updated RAG status.

Green, Amber, Red Tagging

Status	Definition	Criteria
Green	On track	Throughput supports completion. No known blockers. Monte Carlo confidence above 70%.
Amber	At risk but recoverable	Throughput below baseline for more than two days, a blocker actively being worked, or confidence between 40% and 70%. Requires a named recovery action and an owner.
Red	Will not complete without intervention	Blockers open for more than 48 hours, completion confidence below 30%, or strategic context has changed. Requires immediate LGP escalation.

The Amber Trap. The most common DOI failure is Amber items that never become Red because teams fear the visibility of Red. Any Amber item that has not moved to Green within four calendar days automatically converts to Red. Only the Governance Steward can override this, with documented justification.

DOI Health Score

DOI Health Score = (Green Contributions / Total Contributions) × 100

Score	Status	LGP Action
80-100	Strong	Monitor
60-79	Moderate	Review at next LGP
40-59	At Risk	Emergency LGP if Red items present

Score	Status	LGP Action
Below 40	Critical	Escalate to Alignment layer

The trend across sprints is more important than any single score. A score of 75 trending upward is healthier than a score of 80 trending downward.

Red Item Protocol

Step 1: Declaration (within 4 hours). The Delivery Lead updates the Sprint DOI Map and sends a Red Item notification to the Anchor, Governance Steward, and any affected adjacent node Delivery Leads. The notification contains: item name, why it is Red, what has been tried, and what is needed.

Step 2: Initial response (within 4 hours of declaration). If resolvable within the node, it is resolved immediately with a documented decision. If it requires cross-node action, proceed to Step 3.

Step 3: Resolution meeting (within 24 hours). A 30-minute focused session. Output: a named resolution action, a named owner, and a deadline.

Step 4: Closure. Delivery Lead updates the DOI Map and adds a one-sentence note to the sprint retrospective backlog.

Step 5: Cycle review learning. All Red items from the cycle are reviewed at the Cycle Portfolio Review. Patterns feed into the next cycle's risk planning.

Part IV: Ceremonies

Total Ceremony Overhead

In a standard SHIFT sprint for a 5 to 7 person ACN:

Ceremony	Duration	Frequency
AI-First Standup	15 min	Daily (10 days)
Spec Review	45 min	Sprint start
Mid-Sprint DOI Check	30 min	Day 6-7
Sprint Review (incl. IIS Review)	60 min	Sprint end
AI-Augmented Retrospective	60 min	Sprint end
LGP Governance	75 min	Bi-weekly
Cycle Portfolio Review (amortised)	90 min / 6 sprints	Sprint 6

Total per person per sprint: approximately 6.7 hours (8.4% of available capacity). This compares with Scrum at 12 to 16 hours and SAFe at 20+ hours per person per sprint at scale.

Ceremony 1: Spec Review Session

Purpose: Ensure all items entering the sprint meet the SEMI threshold. Identify and resolve specification gaps before they become mid-sprint blockers.

Duration: 45 minutes. **When:** Day 1 of the sprint, or the last hour of the previous sprint's final day.

Facilitation: The Delivery Lead displays the sprint backlog ordered by SEMI score, highest first. For each item scoring 7 or above (Amber band), the team spends no more than 5 minutes on one question: "What is the single highest-risk SEMI dimension in this item, and what action reduces it before work begins?" Actions must be completable before day 3 of the sprint.

Output: All sprint items with confirmed SEMI scores. Amber items with documented mitigation actions. Red items removed and returned to the Anchor with specific gaps identified.

Anti-pattern: Spec Review becoming a second sprint planning session. If scope re-estimation begins, redirect to specification quality only.

Ceremony 2: AI-First Daily Standup

Purpose: Daily coordination, blocker identification, and AI workflow synchronisation.

Duration: 15 minutes. **Format:** Async-first. Written updates in a shared channel; synchronous only when blockers are flagged.

Contributor update format (90 seconds per person): 1. **Progress signal:** "X is 60% complete against criteria Y" or "X is done." Not "I worked on X." 2. **AI workflow note:** "The LLM reviewer flagged three spec gaps; two resolved, one open." Or "No AI tooling involved today." 3. **Blocker flag:** Any blocker. Blockers are addressed after standup, not during it.

After standup (5 minutes, if blockers exist): The Delivery Lead and blocked contributor classify the blocker: internal (resolved by the node), cross-node (escalated to Delivery Lead within 4 hours), or external (escalated to LGP).

Ceremony 3: Mid-Sprint DOI Check-in

Purpose: Reconnect in-progress work to the cycle's strategic objectives. Prevent end-of-sprint alignment surprises.

Duration: 30 minutes. **When:** Day 6 or 7. **Participants:** Delivery Lead, Anchor (mandatory). Contributors for any Amber or Red items.

Fixed agenda: 1. Review Sprint DOI Map RAG status (10 min). Discussion only on items that have changed. 2. Amber review (10 min). Is each recovery action on track? If not, convert to Red. 3. Capacity check (5 min). Can the node complete its committed contributions at current throughput? If not, which contribution is descoped? 4. Named actions with owners and dates (5 min).

Ceremony 4: Sprint Review

Purpose: Demonstrate completed work against acceptance criteria and gather stakeholder feedback.

Duration: 60 minutes (including IIS Review in the final 20 minutes). **When:** Day 10.

Format: First 40 minutes: sprint delivery review. The Anchor opens with a one-sentence summary: was the sprint goal achieved? Each contributor demonstrates their completed work against acceptance criteria. No slide decks. Show the working software, then show the criterion it meets. Stakeholders provide feedback. The Delivery Lead categorises feedback immediately: confirmed acceptance, feedback requiring action, or information.

Final 20 minutes: IIS Review. Learning Cards are presented in under 3 minutes each. Promote, continue, or archive decisions are made in the room.

Anti-pattern: Sprint Review becoming a formal sign-off session. This is a feedback session. Explicitly reframe if it becomes a gate.

Ceremony 5: AI-Augmented Retrospective

Purpose: Systematic learning extraction across three structured tracks that prevent retrospectives from becoming complaint sessions or feel-good exercises.

Duration: 60 minutes. **When:** Sprint ends, same day as Sprint Review or the following morning.

Pre-ceremony (mandatory). Before the retrospective, the AI retrospective agent prepares a synthesis document: throughput trend for the last four sprints, SEMI score distribution for the current sprint, DOI Health Score trend, summary of Learning Cards from the last four IIS cycles, and recurring themes from the last three retrospectives. This is shared one hour before. Reading it before the session eliminates 20 to 30 minutes of context-setting.

Track 1: Delivery System (20 minutes) Opening (5 min, individual silent writing): “What is the single biggest friction point in how we deliver work?” Discussion (10 min): Top one or two systemic delivery issues identified. Actions (5 min): Named actions with owners and

sprint deadlines. These enter the next sprint backlog as technical improvement items, not a parking lot.

Track 2: Collaboration and AI (20 minutes) Opening (5 min, individual silent writing): "Where did AI tooling help us, where did it slow us down, and where did our human collaboration patterns break down?" Discussion (10 min): AI workflow incidents, tooling patterns, cross-node dynamics. Actions (5 min): AI workflow improvements (prompt updates, agent reconfiguration) and collaboration improvements. AI workflow changes are assigned to the AI Responsibility Map owner.

Track 3: Learning and Growth (20 minutes) Opening (5 min, individual silent writing): "What did we learn in this sprint that we should not lose, and what capability are we missing that would make the biggest difference?" Discussion (10 min): IIS insights, skill gaps, pattern library updates, team health signals. This track surfaces important information that is otherwise never said. Actions (5 min): Pattern library updates, IIS theme candidates, capability development items.

Anti-patterns: - "The therapy session": Track 3 without actionable outputs. The AI synthesis document helps prevent this. - "The action graveyard": Actions logged and never revisited. Fix: the first five minutes of every Track 1 opens with "What happened with last sprint's actions?"

Ceremony 6: Cycle Portfolio Review

Purpose: Strategic alignment review at the end of a 12-week cycle. The highest-altitude ceremony in SHIFT.

Duration: 90 minutes. **Participants:** Anchors, Governance Steward, product or strategy leadership.

Agenda: 1. Governance Steward presents the Cycle DOI Summary: Health Score trend, Objective completion, Red item resolution rate. (10 min) 2. Each Anchor presents their Cycle Objective outcomes: what was achieved, what was not. Maximum 5 minutes per objective. No blame. Signal and learning. (30 min) 3. IIS Portfolio Review: themes explored, Learning Cards produced, items promoted, ROI of the 15% IIS capacity allocation. (20 min) 4. Next cycle planning: Cycle Objectives set, IIS themes ratified, ACN formation or dissolution decisions made. (25 min) 5. Closing: next cycle start date confirmed, carry-over items acknowledged. (5 min)

Part V: AI-First Teams

Chapter 9: Spec-Driven Development

AI-first teams operate with a fundamentally different production function. When AI tools can produce a working implementation from a clear specification in hours, the constraint

shifts from coding capacity to specification quality. Vague requirements produce unreliable output regardless of the AI tools involved.

Spec-Driven Development (SDD) treats the specification as the primary engineering artefact. Before any implementation begins, the team produces a complete spec: the problem statement, the acceptance criteria, the edge cases, the integration constraints, and the definition of done. This spec is reviewed, challenged, and signed off before AI-assisted implementation begins.

A complete spec in SHIFT contains: 1. Problem statement: what user or system needs this address? 2. Acceptance criteria: what must be true for this to be considered done? Each criterion must be independently testable. 3. Edge cases: what inputs or states does the system need to handle that are outside the happy path? 4. Integration constraints: what existing systems, schemas, or APIs does this touch? What must not change? 5. AI guidance notes: where is AI output expected to be used, and what review is required? 6. Definition of done: the specific conditions under which this item can move to Done in the SEMI model.

SDD changes sprint planning. Planning is no longer primarily about task decomposition and effort estimation. It is about spec quality review. A sprint is ready to begin when every item has a spec that any contributor, human or AI, could implement without interpretation.

SDD changes retrospectives. The first question is not "why did delivery take longer than expected?" but "where did the spec fail us?" Poor specs are systemic, not personal.

Chapter 10: Team Sizing for AI-First Work

AI-first teams challenge the conventional Agile sizing heuristic of five to nine people. When AI tools multiply individual output, small teams become viable for work that previously required larger groups.

Configuration	Size	When to use	Not suitable for
AI-First Core Team	1-4 people	Well-scoped delivery, clear specs, high AI leverage, understood domain	Discovery work, cross-functional stakeholder alignment, novel domains
Standard ACN	4-8 people	Default configuration for most delivery workstreams	N/A, this is the target size
Coalition ACN	8-15 people	Complex programmes requiring multiple workstreams to maintain coherence	Single-domain work, first adoption cycles

Coalition ACN rules: At 8 to 15 people, the Anchor and Governance Steward become full-time, and the LGP cadence increases. Split into sub-nodes before reaching 15 people.

Chapter 11: The SEMI Model

The SEMI model is SHIFT's estimation and sprint-readiness system. It replaces story points and T-shirt sizes. It is not about predicting exact duration. It is about identifying risk signals before work enters a sprint and creating a consistent language for complexity that improves over time.

SEMI scores four dimensions on a 1 to 3 scale.

S: Specification Quality (1-3)

Score	Label	Definition
1	Clear	Acceptance criteria are written, unambiguous, and testable. Edge cases are documented. A contributor can start without asking clarifying questions.
2	Partial	Acceptance criteria exist but have gaps. A contributor can start but will need one or two clarifications. Some edge cases undocumented.
3	Unclear	Acceptance criteria are missing, vague, or untestable. A contributor cannot start without a significant clarification session.

E: Effort Uncertainty (1-3)

Score	Label	Definition
1	Known	The team has done this before. Similar work completed in the last three cycles. The approach is clear.
2	Similar	Similar to past work but with meaningful differences. Some unknowns exist. The team has a hypothesis but has not validated it.
3	Novel	The team has not done this before. The approach is uncertain. Multiple viable paths may exist.

M: Multi-System Impact (1-3)

Score	Label	Definition
1	Contained	No cross-node dependencies. No external services beyond stable integrations. No schema changes. No security implications.
2	Adjacent	One cross-node dependency or one external service integration. Schema changes within

Score	Label	Definition
		node ownership. Minor security review may be required.
3	Wide	Multiple cross-node dependencies. External integrations with uncertain behaviour. Schema changes affecting other nodes. Security or compliance review required.

I: Implementation Confidence (1-3)

Score	Label	Definition
1	Confident	The implementation approach has been used before in similar contexts. The team is aligned. No significant technical risk.
2	Tentative	The team has a preferred approach but has not validated it. At least one alternative. Some technical risk.
3	Uncertain	No clear implementation approach. A spike may be needed before implementation. High technical risk.

SEMI Composite Score and Sprint Entry Rules

SEMI Score = S + E + M + I (Minimum: 4, Maximum: 12)

SEMI Total	Band	Sprint Entry Rule
4-6	Green: Sprint Ready	Enter the sprint. No additional preparation required.
7-8	Amber: Conditional Entry	Enter only if the Delivery Lead and Anchor agree on a documented mitigation for the highest-scoring dimension.
9-10	Red: Spec Required	Cannot enter the sprint. Return to the Anchor for specification improvement.
11-12	Black: Decompose	Items are too large or complex. Decompose into child items and re-score each.

AI-Specific Scoring Modifiers

Condition	Dimension	Modifier
AI output is non-deterministic and criteria do not account for variance	S	+1
AI model is externally hosted and rate-limited	M	+1

Condition	Dimension	Modifier
AI model requires prompt engineering not yet documented	I	+1
AI output is the primary user-facing output	E	+1
AI agent has cross-system tool access	M	+1 per additional tool beyond 2

Modifiers are additive but capped: no single dimension exceeds 3.

SEMI Pattern Analysis

SEMI scores across sprints and cycles produce a pattern library that improves estimation quality over time. The Delivery Lead reviews patterns at each cycle retrospective.

Pattern	Systemic Signal	Fix
Consistent S=3	Specs are written too late or by people disconnected from implementation	Introduce Spec Review earlier, pair Anchor with contributor on spec writing
Consistent E=3 on one work type	Team treats familiar work as novel, not building pattern familiarity	Document an implementation pattern for this type; E should decrease to 1 or 2 after
Consistent M=3	Node domain boundaries are too wide, or cross-system work proceeds without interface agreements	Tighten capability domain, establish formal interface documents
Consistent I=3	Team lacks confidence in implementation approaches	Targeted IIS themes on technical capability, pair contributors on complex items

SEMI Scoring Reference Table

Work Type	S	E	M	I	Total	Notes
Bug fix, known root cause	1	1	1	1	4	Sprint ready
Bug fix, unknown root cause	2	3	1	2	8	Conditional: timebox investigation
New UI component (standard)	1	1-2	1	1	4-5	Sprint ready
New API endpoint (standard)	1-2	1-2	1-2	1	4-7	Usually sprint ready

Work Type	S	E	M	I	Total	Notes
New API endpoint (external auth)	2	2	3	2	9	Spec required: clarify auth
LLM integration (new)	3	3	2	3	11	Decompose: separate spike from integration
LLM integration (established pattern)	1	2	2	1	6	Sprint ready after pattern documented
Data migration (small)	1	2	2	1	6	Sprint ready with rollback plan
Data migration (large, cross-system)	2	3	3	2	10	Red: full spec including rollback required
Infrastructure change (proven)	1	1	1	1	4	Sprint ready
Security or compliance feature	2-3	2	3	2	9-10	Spec required, compliance review mandatory

---

Chapter 12: Forecasting with Throughput and Monte Carlo

Velocity-based forecasting assumes stable team composition, consistent story sizing, and a predictable production function. AI-first teams have none of these properties. A team's effective throughput can double between sprints as prompting skills improve or new tooling is adopted. Velocity charts in this context are misleading.

SHIFT uses **throughput forecasting**: counting the number of work items completed per sprint, regardless of estimated size. Throughput is more stable than velocity because it is not distorted by estimation inflation or by the wide variance in how AI tools handle different item types.

Throughput vs. Velocity

Dimension	Velocity (story points)	Throughput (items)
Unit consistency	Weak (points vary by estimator)	Strong (item = item)
AI work compatibility	Poor	Moderate, calibratable with SEMI bands
Gaming risk	High (point inflation)	Low
Stakeholder clarity	Low	High

What Counts as a Completed Item

An item is complete when: it meets all its acceptance criteria, it has passed its required review step (including human review for AI-assisted outputs), and it is deployed to at least a staging environment or integration-tested.

Items “in review” that have not received a confirmed pass do not count. AI-generated outputs that have not received human review do not count. “Soft done” items moved to Done without meeting criteria do not count.

Monte Carlo Forecasting

Given a set of remaining items and a historical throughput distribution, Monte Carlo simulation runs thousands of randomised completions to produce a probability distribution of completion dates.

SHIFT probability bands:

Band	Probability	Use for
P50	50%	Internal planning only
P70	70%	Sprint goal setting, internal commitment
P85	85%	Stakeholder commitments
P95	95%	Contractual or external commitments

Stakeholder communication template: “Based on our last 10 sprints, we have 85% confidence that the authentication feature will be complete by 14 March. If our throughput holds to our recent patterns, we expect to finish by 10 March.”

Minimum data requirement: 8 sprints of throughput data before Monte Carlo is reliable from internal data. Before that, use reference class forecasting.

Reference Class for Early Baseline (Sprints 1 Through 7)

Use industry-baseline throughput distributions until internal data is available. Blend reference class data with actual throughput data from sprint 3 onwards (50/50 blend). By sprint 8, actual data dominates.

Team Size	Green Items/Sprint (reference)	Amber Items/Sprint (reference)
3 people	5-8	2-4
5 people	8-13	3-6
7 people	12-18	5-9

Always flag to stakeholders when reference class data is in use: “This forecast uses industry baseline data. Treat P85 as P70 until sprint 8.”

Chapter 13: AI-First Practices

Prompt Library. A node-maintained library of effective prompts for common task types: spec writing, code generation, test generation, review, documentation. Maintained by Node Contributors. Referenced before starting AI-assisted work. Updated whenever a prompt

produces significantly better or worse results than expected. The Prompt Library is a first-class team artefact, not an individual's notes.

AI Review Protocol. A structured checklist for reviewing AI-generated output: Does the output match the spec acceptance criteria? Does it handle the documented edge cases? Does it behave correctly in integration? Are there security implications? Has it been tested against the definition of done? No item transitions to Done without this review being documented as completed.

Pair-with-AI approach. AI is a collaborator, not an autonomous agent. The contributor owns the decision on all AI output. Pairing means the contributor is actively shaping the AI's work: writing the spec, directing the prompts, reviewing the outputs, and deciding what to accept, modify, or reject. Pair-with-AI is not "run the AI and accept the output."

AI Retro Track. Track 2 of the AI-Augmented Retrospective (Collaboration and AI) is specifically designed to surface AI tool effectiveness. Each sprint, the team reviews: which prompts in the Prompt Library are still working? Which needs updating? Are there new prompt patterns worth capturing? This track is the maintenance mechanism for the team's AI tooling capability.

Chapter 14: AI-First Mindset Shifts

From estimation to specification quality. Time spent debating story point estimates is time not spent clarifying what needs to be built. A good spec is worth more than an accurate estimate. Invest in specification quality; let throughput data handle forecasting.

From velocity to throughput. Velocity measures effort-weighted output. In teams where AI can produce ten times the output on a well-specified item versus a vague one, effort weighting becomes noise. Count items completed. Track the distribution. Use Monte Carlo for forecasting.

From heroics to system quality. AI tools make the quality of the specification, the tooling, and the review process the binding constraint, not the individual. Team-level system quality replaces individual brilliance as the primary performance lever.

From done to verified. AI-generated output needs rigorous verification. The definition of done must include explicit review steps: does the output match the spec? Does it pass the documented edge cases? Does it behave correctly in integration? Done means verified.

From synchronous planning to asynchronous alignment. If the spec is clear, most planning questions resolve before the meeting. AI-first teams do their best alignment work asynchronously, leaving synchronous time for spec ambiguity resolution, DOI escalations, and retrospective inquiry.

From prompting to spec-then-prompt. The quality of an AI prompt is bounded by the quality of the spec behind it. A vague spec produces a vague prompt produces unreliable output. Invest in the spec first; the prompt follows naturally.

From AI as a tool to AI as a collaborator. The distinction is accountability. A tool is used. A collaborator is directed, reviewed, and held to a standard. Node Contributors direct AI collaborators, review their outputs, and own the results.

Part VI: Integration

Chapter 15: SHIFT + Scrum

Scrum and SHIFT are the most natural integration. SHIFT's sprint structure is derived from Scrum. The differences are in ceremony design, estimation, and the addition of AI-first, IIS, DOI, and governance layers.

What SHIFT keeps from Scrum: - 2-week sprint cadence and sprint goal concept. - Sprint Review as the primary feedback ceremony. - Retrospective structure (SHIFT augments it significantly).

What SHIFT replaces: - Story points with SEMI scoring and throughput tracking. - Velocity with Monte Carlo forecasting. - Product Owner with the Anchor role (broader outcome accountability, no single backlog ownership concentration). - Scrum Master with the Delivery Lead (operational, not coaching). - Backlog Refinement with the Spec Review ceremony. - Standard Retrospective with the 3-track AI-Augmented Retrospective.

What SHIFT adds: - DOI model (not present in Scrum). - IIS (not present in Scrum). - AI-first Standup format. - SEMI scoring. - LGP (replaces informal governance or management escalation). - Monte Carlo forecasting.

Migration path:

Phase	Sprints	Change
1	1	Add SEMI scoring to backlog refinement. Introduce AI-first standup format.
2	2	Replace story points with throughput tracking.
3	3	Introduce IIS at 10% capacity. Replace standard retrospective with 3-track format.
4	4	Introducing the DOI model. Map sprint contributions to cycle objectives.
5	6	Introduce Spec Review ceremony. Retire backlog refinement.
6	8	Begin Monte Carlo forecasting. Retire velocity.
Full	Cycle 2	Full SHIFT operation.

Chapter 16: SHIFT + Kanban

Kanban and SHIFT share flow-based thinking. The integration adds time-boxing and strategic alignment to Kanban's continuous flow without destroying its flow-based insights.

What SHIFT keeps from Kanban: WIP limits (SHIFT endorses WIP limits at node level), flow metrics (cycle time, lead time, throughput, directly compatible with SHIFT's throughput model), and visualisation discipline.

What SHIFT adds: Sprint time-boxing as a planning and review cadence (not a flow constraint), the DOI model, IIS as a dedicated innovation swim lane, and Spec Review applying SEMI before items enter the WIP queue.

SHIFT-Kanban board design: - Columns: To Do, Spec (SEMI scoring), In Progress, Review (AI review step), Done. - A dedicated Innovation swim lane (IIS items, separate WIP limit). - DOI tags visible on all cards (Green/Amber/Red indicator). - SEMI scores on card faces.

Transition framing for Kanban teams: "The sprint is a measurement and alignment cadence, not a flow constraint. Your flow continues uninterrupted. The sprint provides the review points and DOI alignment moments that Kanban does not define."

Flow metric integration: Kanban cycle time data is an input to SEMI E-dimension calibration. This creates a feedback loop: Kanban flow data improves SEMI calibration, which improves sprint planning quality.

Chapter 17: SHIFT + SAFe (approximately 40% ceremony reduction)

SAFe is the most common enterprise Agile framework and the most ceremony-heavy. SHIFT can reduce SAFe ceremony overhead by approximately 40% while preserving or improving alignment outcomes.

SAFe ceremony replacement table:

SAFe Ceremony	SAFe Duration	SHIFT Replacement	SHIFT Duration	Reduction
Iteration Planning	4 hours	Sprint Planning + Spec Review	90 min + 45 min	67%
Daily Scrum	15 min	AI-First Standup	15 min	0% (same)
Iteration Review	60 min	Sprint Review	60 min	0% (deeper)
Iteration Retrospective	60 min	AI-Augmented Retro	60 min	0% (deeper)
Backlog Refinement	2 hours	Spec Review (SEMI-driven)	45 min	63%
PO Sync	30 min weekly	DOI async update	15 min	50%

SAFe Ceremony	SAFe Duration	SHIFT Replacement	SHIFT Duration	Reduction
Scrum of Scrums	30-60 min weekly	LGP (bi-weekly)	75 min bi-weekly	38%
ART Sync	60 min bi-weekly	Merged into LGP	Absorbed	100%

Net reduction per person per sprint: approximately 3.5 to 5 hours. Over a 12-week PI, this saves 21 to 30 person-hours per contributor.

SAFe role mapping:

SAFe Role	SHIFT Role
Product Owner	Anchor
Scrum Master / RTE	Delivery Lead + Governance Steward (split by layer)
System Architect	Senior Node Contributor (AI Responsibility Map owner)
Product Manager	Governance Steward (strategy alignment)

Chapter 18: SHIFT + LeSS

LeSS and SHIFT share a foundational philosophy: scaling should be achieved by descaling. LeSS deliberately removes coordination roles, which can create a vacuum in organisations that are not fully ready for that transition.

What SHIFT adds to LeSS: - ACN structure adds outcome ownership to LeSS feature teams. - LGPs provide lightweight coordination where LeSS removes it, without adding the overhead LeSS was designed to avoid. - DOI distributes strategy alignment to sprint level, complementing the overall Product Owner connection. - IIS provides the structured innovation capacity that LeSS does not define.

Where LeSS and SHIFT diverge: LeSS uses a single Product Backlog owned by a single Product Owner. SHIFT distributes ownership across Anchors by capability domain. In a LeSS + SHIFT integration, the Product Owner becomes a strategic Anchor for the overall product, with ACN Anchors as domain sub-owners.

SHIFT-LeSS transition value: SHIFT + LeSS works particularly well for organisations transitioning from SAFe who want the lean philosophy of LeSS but still need some coordination structure during the transition period. SHIFT provides that bridge.

Chapter 19: SHIFT + OKRs

OKRs are a goal-setting framework, not a delivery framework. SHIFT's DOI model is specifically designed to bridge between OKRs and sprint delivery.

The fundamental OKR problem DOI solves: OKRs are set quarterly. Sprint delivery happens fortnightly. There is no standard mechanism in OKR frameworks for checking, mid-sprint, whether the work being done is actually moving the OKR forward. Teams often discover at quarter end that they delivered excellent software with no measurable impact on the stated Objective. DOI makes OKR contributions visible and checkable every sprint.

OKR to SHIFT mapping:

OKR Level	SHIFT Equivalent	Cadence
Company OKR	Alignment layer input	Quarterly
Team OKR	Cycle Objective (in DOI map)	12-week cycle
Key Result	DOI Sprint Contribution outcome signal	Sprint
Initiative / Output	Sprint Contribution (delivery work)	Sprint

DOI Health Score as leading OKR indicator: A declining DOI Health Score mid-cycle is an early signal that the team may not hit its Key Results. This surfaces the problem six weeks before quarter end rather than at the quarterly review.

Integrated OKR grading: SHIFT does not use standard 0 to 1.0 OKR scoring. It uses DOI Health Score trend plus a qualitative outcome assessment at cycle end. This prevents the gaming that OKR scoring often produces, by connecting the assessment to delivery evidence.

Part VII: Adoption

Chapter 20: SHIFT Maturity Model

The SHIFT Maturity Model is a navigation tool, not a certification programme. Teams use it to understand where they are, what to focus on next, and what good looks like at each level.

Level 1: Established Foundation

The team has adopted SHIFT's core delivery mechanics. Ceremonies are running, SEMI scoring is in use, throughput is being tracked, and the basic ACN structure is in place.

Self-assessment questions: 1. Does every sprint end with a Sprint Review where working software is demonstrated against acceptance criteria? 2. Does every sprint item have a SEMI score before entering the sprint? 3. Are you tracking the number of items completed per sprint (throughput)? 4. Can every team member describe the node's capability domain in one sentence? 5. Does every Node Contributor own the review step for their AI-assisted outputs?

Advancement to Level 2: 4 of 5 questions yes, consistently across at least 3 sprints.

Level 2: Aligned Delivery

The team is operating DOI alignment, IIS is running, and governance is functioning through the LGP.

Self-assessment questions: 1. Can you point to the Cycle Objective that each current sprint item contributes to? 2. Have you had at least one Red DOI item this cycle and resolved it? 3. Is IIS happening every cycle, and do Learning Cards exist as output? 4. Is the LGP pre-read being distributed 24 hours before every session? 5. Is throughput data from the last 6+ sprints available and in use for planning?

Advancement to Level 3: 4 of 5 questions yes, consistently across at least one full cycle (12 weeks).

Level 3: Adaptive Intelligence

The team uses SHIFT's feedback loops to improve its delivery system continuously. Monte Carlo forecasting is in use, the pattern library is active, and IIS is producing promoted outputs.

Self-assessment questions: 1. Are you communicating forecast confidence using probability bands (P70/P85) to stakeholders? 2. Has at least one IIS theme been promoted through the full funnel to a cycle investment? 3. Does the team consult the pattern library before scoring SEMI dimensions? 4. Is the AI retrospective synthesis pulling data from at least four previous sprints? 5. Are retrospective actions being completed at a rate above 70%?

Advancement to Level 4: 4 of 5 questions yes, consistently across two consecutive cycles.

Level 4: System Contribution

The team is a net contributor to the SHIFT ecosystem. They generate pattern library content that benefits other teams, contribute to SEMI calibration improvements, and operate as a reference team.

Self-assessment questions: 1. Have you shared pattern library content with at least two other teams in the last cycle? 2. Are AI agent roles from your node being used or adapted by other nodes? 3. Is your SEMI calibration accurate enough that you rarely have surprise scope overruns? 4. Can a new team member contribute meaningfully within 5 days, using documented materials? 5. Has at least one IIS output from your team influenced a product roadmap or cycle portfolio decision?

Using the Maturity Model

The self-assessment is run at cycle end, facilitated by the Governance Steward. It takes 30 minutes. It is not a scoring exercise to be reported upward. It is a navigation exercise: what should the team focus on in the next cycle?

The maturity model is not a race. A 3-person team at Level 2 is operating excellently. A 15-person team at Level 1 is struggling and should focus on the fundamentals.

Chapter 21: Getting Started

Phase 0: Ground Zero (Week Before Sprint 1)

Objective: Establish the minimum viable structure to begin Sprint 1.

Day 1-2: Team diagnostic. Run a 90-minute session. Map: current ceremonies, current pain points, current AI tool usage. Document honestly.

Day 2-3: Role assignment. Identify Anchor, Delivery Lead, and Node Contributors. Document role combinations for small teams.

Day 3-4: Node definition. Write the capability domain in one sentence. Write the initial AI Responsibility Map: for each contributor, what tasks are AI-assisted, AI-led with review, or human-only?

Day 4-5: First backlog preparation. Score the top 20 backlog items using SEMI. The goal is not accuracy. The goal is to surface the team's disagreements about specification quality and complexity.

Day 5: Sprint 1 planning. Select items scoring 6 or below for Sprint 1. Items scoring 7 or above go to a "to be specced" list for the Anchor.

Phase 1: Foundation Sprints (Sprints 1-3)

Sprint 1: Introduce AI-first standup, throughput tracking, Sprint Review, and 3-track retrospective. Do not yet introduce the DOI model, IIS, Monte Carlo, or LGP. Retrospective focus: Was SEMI calibration right? What did we miss?

Sprint 2: Introduce Spec Review ceremony. Refine SEMI calibration. Start throughput distribution tracking. Retrospective focus: Is the Spec Review changing spec quality?

Sprint 3: Introduce IIS at 10% capacity. One theme, one Learning Card. Introducing AI Responsibility Map review. Retrospective focus: How did IIS feel? Did the Learning Card capture something real?

Phase 1 milestone: Maturity Level 1.

Phase 2: Alignment Layer (Sprints 4-6)

Sprint 4: Introduce Cycle Objectives and DOI map. Run first mid-sprint DOI check.

Sprint 5: Introduce LGP. Run the first bi-weekly governance session with a simplified pre-read. Introducing the Red Item Protocol. Increase IIS to 15%.

Sprint 6: Run first Cycle Portfolio Review. Calculate first DOI Health Score. AI synthesis in retrospective for the first time.

Phase 2 milestone: Maturity Level 2.

Phase 3: Intelligence Layer (Sprints 7-9)

Sprint 7: Introduce Monte Carlo forecasting using reference class data. Communicate the first P70/P85 forecast to a stakeholder. Start pattern library with five entries.

Sprint 8: Retire reference class data (8 sprints of internal data now available). Refine SEMI scoring. Evaluate IIS Learning Cards for promotion to pilot.

Sprint 9: Full SHIFT operation at Maturity Level 3 target. Review the last four retrospective syntheses for recurring unresolved themes.

Phase 3 milestone: Maturity Level 3.

Phase 4: Full Operation (Sprints 10-12+)

Monte Carlo from internal data. Pattern library contributions to other nodes. Maturity Level 4 self-assessment. If other teams are adopting SHIFT, this team operates as a reference.

Getting Started Summary

Phase	Sprints	Key Introductions	Maturity Target
Phase 0	Week -1	Roles, domain, SEMI backlog	Pre-Level 1
Phase 1	1-3	Standup, throughput, ceremonies, Spec Review, IIS 10%	Level 1
Phase 2	4-6	DOI map, LGP, Cycle Objectives, Red Item Protocol, IIS 15%	Level 2
Phase 3	7-9	Monte Carlo, pattern library, IIS pilot promotion	Level 3
Phase 4	10-12+	Full calibration, cross-team contribution	Level 3-4

Appendix A: Core Metrics Quick Reference

Metric	What it measures	Frequency	Owner
Throughput (items/sprint)	Delivery rate	Daily	Delivery Lead
SEMI Score (4-12)	Item readiness and risk	At sprint entry	Delivery Lead + Contributors
DOI Health Score (0-100)	Strategy-to-sprint alignment	Sprint start, mid-sprint, sprint end	Delivery Lead
IIS Capacity (%)	Innovation investment	Sprint planning	Anchor
Monte Carlo P70/P85	Delivery confidence	Sprint end	Delivery Lead

---

Appendix B: SHIFT Vocabulary

Term	Definition
ACN	Adaptive Collaboration Node: primary delivery unit, owns a capability domain
LGP	Lean Governance Pod: bi-weekly cross-node decision forum, maximum 75 minutes
IIS	Innovation-Integrated Sprint: ring-fenced exploration capacity (15% standard) within every sprint cycle
DOI	Dynamic Objectives Integration: mechanism connecting sprint work to cycle objectives via RAG tagging
SEMI	Specification Quality, Effort Uncertainty, Multi-system Impact, Implementation Confidence: the four-dimension item readiness model
Anchor	Accountable owner of an ACN's capability domain and outcomes
Delivery Lead	Operational owner of an ACN's sprint delivery system
Governance Steward	Owner of LGP function and governance system health
Node Health Card	One-page ACN status artefact updated every sprint
Learning Card	Structured IIS output: what was explored, what was learned, what is recommended
Cycle	12-week delivery and alignment period (six sprints)
P70 / P85	Monte Carlo probability bands: 70% and 85% confidence delivery forecasts
DOI Health Score	Aggregate cycle-level alignment score based on RAG Sprint Contributions
Red Item Protocol	Structured escalation process for Red DOI items: 4-hour declaration SLA, 24-hour resolution meeting
AI Responsibility Map	Node-level document defining which tasks are AI-assisted, AI-led with review, or human-only
Spec-Driven Development (SDD)	Practice treating the specification as the primary engineering artefact before any implementation begins
Pattern Library	Shared repository of implementation patterns, SEMI calibration insights, and IIS learnings
Throughput	Number of work items completed per sprint, the primary SHIFT delivery metric

Appendix C: Frequently Asked Questions

Starting SHIFT

Q: We want to start implementing SHIFT on our team today. What are the three most important things to do first?

A: In order of priority:

1. **Define your ACN clearly.** Write the capability domain in a single sentence. If it takes two sentences, the scope is too broad. This is the most common first failure: teams start SHIFT ceremonies before they have agreed on what the team actually owns. Everything else depends on this.

2. **Score your backlog with SEMI.** Take the top 15 to 20 items in your backlog and score them using the four dimensions (Specification quality, Effort uncertainty, Multi-system impact, Implementation confidence). Do not aim for precision in the first round. Aim for surfacing disagreement: when two people give the same item very different scores, that is the conversation that needs to happen before the work begins.

3. **Start tracking throughput from sprint one.** Count the number of items marked Done at the end of each sprint. Not story points, not hours, not effort. Items completed and meeting acceptance criteria. Even with no other SHIFT elements in place, eight sprints of throughput data will transform your planning conversations.

Everything else — IIS, DOI, LGP, Monte Carlo — layers on top of these three. Teams that skip the foundation and go straight to the advanced mechanisms fail consistently.

Q: We are already running Scrum. What changes in sprint 1 if we adopt SHIFT?

A: In sprint 1, make exactly three changes. Do not change anything else.

1. **Replace standup format.** Move from "what did I do, what will I do, any blockers" to three signals: a progress signal against a measurable criterion, an AI workflow note (even if the note is "no AI involvement today"), and a blocker flag. The blocker is addressed after standup, not during it.

2. **Add SEMI scoring to refinement.** Before any item enters the sprint, it gets a SEMI score. Items scoring 9 or above do not enter the sprint. Items scoring 7 to 8 enter only with a named mitigation on the highest-risk dimension. This alone will visibly reduce mid-sprint surprises.

3. **Replace your retrospective format with the 3-track structure.** Track 1: delivery system. Track 2: collaboration and AI. Track 3: learning and growth. Do not combine tracks or skip one because "we don't have AI issues yet." Track 2 surfaces human collaboration patterns even when AI tools are not involved.

Everything else in sprint 1 stays the same. IIS, DOI, LGP, and Monte Carlo are introduced in subsequent sprints per the phased adoption plan in Chapter 21.

Q: What is the absolute minimum version of SHIFT we can run with a 3-person team?

A: With three people, SHIFT collapses to its core:

- **Roles:** One person is the Anchor and Governance Steward combined. One person is the Delivery Lead and primary contributor combined. One person is a contributor.
- **Ceremonies:** AI-first standup daily, Spec Review at sprint start, Sprint Review, 3-track Retrospective. Drop IIS (IIS requires at least 4 people to be meaningful), drop LGP (replace with a weekly 30-minute alignment meeting with one external stakeholder or product owner), drop the formal Cycle Portfolio Review (replace with a 45-minute conversation at the end of each 12-week cycle).
- **Metrics:** Throughput only. SEMI scoring. Skip Monte Carlo until you have 8 sprints of data.
- **DOI:** Maintain a simplified DOI map: one or two cycle objectives, each with current sprint contributions tagged Green, Amber, or Red. Review at the mid-sprint check.

A 3-person team running this minimum version is operating SHIFT. They are at the pre-Level 1 to Level 1 boundary and should focus on getting SEMI calibration right before adding anything else.

Q: How long before we see results from adopting SHIFT?

A: Realistic expectations by phase:

- **Sprints 1-2:** You will see process friction increase temporarily. Teams naming spec gaps they previously ignored, standup taking slightly longer as people adjust to the new format, SEMI scores causing debates. This is not failure. This is the framework surfacing problems that already existed invisibly.

- **Sprints 3-4:** SEMI calibration begins to stabilise. Mid-sprint surprises decrease noticeably. Teams report that planning conversations are more honest and shorter.

- **Sprint 6-8 (first full cycle end):** Throughput data becomes reliable enough to use for planning conversations. The Cycle Portfolio Review is the first moment most teams feel the framework operating at altitude, connecting their sprint work to strategic objectives in a way that is evidenced, not asserted.

- **Cycle 2 onwards:** IIS begins producing Learning Cards worth promoting. Monte Carlo forecasts are based on internal data and carry credibility with stakeholders. DOI health scores give the team and leadership a shared, objective view of alignment.

The failure mode to avoid: abandoning SHIFT in sprint 2 or 3 because it feels like overhead. The overhead in sprints 1 through 3 is the investment period. Teams that stay with it past sprint 4 uniformly report that the overhead inverts into a net reduction in wasted coordination time.

Q: Our organisation will not let us drop story points. Can we run SHIFT alongside story point estimation?

A: Yes, with a clear separation of purpose. Use story points exclusively for external reporting or contractual commitments where they are required. Use SEMI and throughput for internal planning, sprint readiness, and forecasting. Do not mix them in the same conversation.

The practical approach: score SEMI, run the sprint, track throughput, and run Monte Carlo forecasting from throughput data. If someone external asks for a story point count, derive it from throughput after the sprint using your historical average points per item. This keeps SEMI and throughput clean while satisfying reporting requirements that you cannot control.

Warn your team explicitly: SEMI scores and story points answer different questions. SEMI answers "is this item ready to enter the sprint and how complex is it?" Story points attempt to answer "how much effort will this take?" These are not the same question, and conflating them is the most common estimation failure in Agile teams.

Metrics and Tracking

Q: What are the core SHIFT metrics and what exactly do I track for each one?

A: SHIFT uses five core metrics. Here is exactly what to track for each:

Throughput (items per sprint) Track: the count of work items moved to Done in each sprint that meet all acceptance criteria, have passed their review step (including AI review if applicable), and have been deployed to at least staging. Record as a single number per sprint. Build a rolling 10-sprint dataset. From sprint 8 onwards, this dataset drives Monte Carlo forecasting. What not to track as throughput: items partially done, items in review but not confirmed, items moved to Done informally without meeting criteria.

SEMI Score (4-12 per item) Track: the SEMI score for every item at sprint entry. Log the four individual dimension scores, not just the composite. After each sprint, review: which dimension scores were wrong? Where did the team underestimate M=1 when it should have been M=3? This calibration exercise is how SEMI improves over time. Aggregate SEMI scores by dimension across a cycle to surface systemic patterns (consistent S=3 means specs are being written too late; consistent I=3 means technical confidence is low across the board).

DOI Health Score (0-100) Track: at sprint start, mid-sprint, and sprint end. Count the number of Green-tagged Sprint Contributions divided by total Sprint Contributions, multiplied by 100. Record the score and the trend. A single low score is less important than a declining trend. The score is useless if it is not updated at the mid-sprint check, which is the moment it is most actionable.

IIS Capacity (%) Track: at sprint planning. The percentage of total sprint capacity allocated to IIS. The standard is 15%. Record whether IIS ran, what theme was explored, how many Learning Cards were produced, and whether any were promoted. This creates an IIS portfolio view across cycles.

Monte Carlo P70/P85 Track: at sprint end, from sprint 8 onwards. For any active work thread with more than three items remaining, run a Monte Carlo simulation using the last 8-sprint throughput distribution. Record the P70 date (internal planning) and P85 date (stakeholder commitment). Compare actuals against P85 forecasts over time: if actuals consistently fall outside P85, your throughput distribution has a systematic bias to investigate.

Q: How do we calculate the DOI Health Score in practice, sprint by sprint?

A: The calculation is simple. The discipline is in the tagging.

At sprint planning, every item in the sprint backlog is tagged against a Cycle Objective as Green, Amber, or Red. Most items will be Green at planning. The Delivery Lead updates tags at the mid-sprint check (Day 6-7) and at sprint end.

Formula: DOI Health Score = (count of Green-tagged Sprint Contributions ÷ total Sprint Contributions) × 100.

Example: Sprint has 12 Sprint Contributions. At mid-sprint check, 8 are Green, 3 are Amber, 1 is Red. DOI Health Score = (8 ÷ 12) × 100 = 67. This score is Moderate and triggers a review at the next LGP. The 1 Red item triggers the Red Item Protocol immediately.

The common mistake: teams tag items Green at planning and never update them. A DOI Health Score of 100 at sprint end that was not updated mid-sprint is not a health score. It is fiction. The value of DOI is the mid-sprint signal, not the sprint-end summary.

Q: How do we run Monte Carlo in practice? Do we need special software?

A: You do not need special software, though dedicated tools make it significantly easier.

Manual approach (workable for early sprints): Record throughput per sprint in a spreadsheet (items completed per sprint). To simulate a Monte Carlo run manually, roll dice or use a random number generator to sample throughput values from your historical data. Repeat 500 times, recording how many simulations complete a backlog of N items within K

sprints. The percentage of simulations completing within each timeframe gives you your probability bands. This is tedious but teaches the underlying logic.

Spreadsheet approach: Use a spreadsheet with a throughput history column, a RANDBETWEEN or random sample formula to simulate draws, and a count of remaining items. Set up 1,000 simulation rows. The P50, P70, P85, and P95 outcomes emerge from the distribution of results. Free templates for this approach exist and are easy to adapt.

Dedicated tools (recommended from sprint 8 onwards): - **ActionableAgile Analytics** (actionableagile.com): the most rigorous Monte Carlo tool for Agile teams. Connects directly to Jira and Azure DevOps. Produces throughput run charts, cycle time scatterplots, and Monte Carlo "How Many" and "When" simulations out of the box. This is the recommended tool for SHIFT teams. - **Nave** (nave.app): similar to ActionableAgile, strong integration with Jira. Good visualisation of flow metrics alongside Monte Carlo. - **Focusedlabs Throughput Forecaster** (free spreadsheet): downloadable Google Sheets template, no software subscription required, good for teams running 3 to 5 nodes. - **Azure DevOps Analytics**: built-in throughput and cycle time reports. Monte Carlo requires a third-party extension (ActionableAgile for ADO) or a connected spreadsheet.

The minimum viable setup: a shared spreadsheet with one row per sprint, one column for items completed. This takes five minutes to maintain per sprint and provides everything needed for manual P70/P85 calculations.

Q: What leading indicators should we track, not just lagging ones?

A: SHIFT distinguishes between lagging indicators (what happened) and leading indicators (what is about to happen). Most Agile teams track only lagging indicators and are surprised by outcomes they could have seen coming.

Leading indicators in SHIFT:

- **SEMI score distribution at sprint entry:** A sprint where 40% of items entered with Amber or Red scores is a leading indicator of mid-sprint blockers and incomplete work at sprint end. Check this at the Spec Review, before delivery begins.
- **Amber item age in the DOI map:** An Amber item on day 3 of a sprint that has not moved to Green by day 5 will almost certainly become Red. Track Amber item age daily, not just at the mid-sprint check.
- **Throughput trend (last 3 sprints):** A declining trend across three consecutive sprints is a leading indicator of a systemic delivery problem. A single low-throughput sprint is noise. Three consecutive is a signal. Do not wait for the cycle retrospective to investigate.
- **IIS Learning Card signals:** A cycle where all IIS Learning Cards are Neutral or Negative is a leading indicator that themes are not well-chosen or that IIS capacity is

too low to produce meaningful signals. Investigate theme selection at the next IIS Review.

- **Spec Review duration:** If Spec Review consistently runs over 45 minutes, items are arriving at sprint planning without adequate specification. This will manifest as mid-sprint blockers. The Spec Review duration is a leading indicator of sprint predictability.

Tooling: Jira, Azure DevOps, and Others

Q: We use Jira. How do we configure it to support SHIFT without a major overhaul?

A: SHIFT maps well to Jira with five targeted configuration changes. None require a new project structure.

1. SEMI Score field Add a custom field to your issue type: Number field named "SEMI Score" (range 1-12). Add four additional custom number fields for S, E, M, and I individually (range 1-3 each). Display all five on the issue, create and edit screens. This is the single most impactful configuration change: it forces SEMI scoring to be done at the time of issue creation or refinement, not as a separate process.

2. DOI Status field Add a custom field: Single-select with options Green, Amber, Red, and Not Tagged. Add this field to the sprint board card face so it is visible at a glance. Update this field at mid-sprint. Create a Jira dashboard gadget filtered by sprint showing the count of each DOI status. This gives you the DOI Health Score calculation at a glance.

3. IIS label or Epic Create a dedicated Epic named "IIS [Cycle Number]" for each cycle's IIS work. Alternatively, use a label "IIS" on all IIS-related issues. This lets you filter IIS throughput from delivery throughput separately, which is important: IIS items should not be included in your delivery throughput baseline used for Monte Carlo.

4. Throughput tracking Use the Jira Velocity Chart in item count mode, not story point mode. Navigate to your board, click Reports, select Velocity Chart, and switch the metric from Story Points to Issue Count. This gives you throughput per sprint. Export this data to a spreadsheet every sprint end and maintain your own rolling 10-sprint throughput dataset. Do not rely solely on Jira's built-in velocity chart for Monte Carlo inputs, as it aggregates in ways that can obscure the raw distribution you need.

5. ActionableAgile integration Install the ActionableAgile Analytics plugin from the Atlassian Marketplace. Connect it to your Jira project. This adds a Monte Carlo simulator, throughput run chart, cycle time scatterplot, and WIP ageing chart directly inside Jira. From sprint 8 onwards, run your P70/P85 forecasts from the ActionableAgile "When" simulation. This replaces the need for spreadsheet-based Monte Carlo.

Node Health Card: Maintained as a Confluence page linked from the Jira project. Update the four fields (throughput trend, DOI Health Score, IIS status, open dependencies) at sprint

end. The Node Health Card is a human-readable summary, not a Jira dashboard — keep it in Confluence or a similar doc tool.

Q: We use Azure DevOps. How do we configure it for SHIFT?

A: Azure DevOps has strong built-in analytics that align well with SHIFT, particularly for throughput tracking. The configuration is similar in intent to Jira but uses ADO's terminology.

1. SEMI Score fields In Process customisation (Organisation Settings > Boards > Process), add custom fields to your work item type (usually User Story or Product Backlog Item): one integer field named "SEMI Score" and four integer fields named "S-Score", "E-Score", "M-Score", "I-Score". Add these to the work item form under a "SHIFT" section. Fields are visible on the backlog and can be filtered in queries.

2. DOI Status field Add a custom picklist field named "DOI Status" with values: Green, Amber, Red, Not Tagged. Add it to the work item form. Create a query in Azure DevOps Boards that groups by DOI Status filtered to the current sprint. Pin this query as a board widget on your team dashboard. Update the DOI Status field at mid-sprint and sprint end.

3. IIS work items Create a dedicated Tag "IIS" for all innovation sprint work items. Alternatively, create an Area Path or Feature named "IIS" to group all IIS work. Use this to separate IIS throughput from delivery throughput in analytics. In the Analytics tab, you can filter by area path or tag to produce a separate throughput chart for IIS items.

4. Throughput tracking with ADO Analytics Navigate to Boards > Analytics > Throughput. ADO provides a built-in throughput report showing items completed per sprint. Switch the granularity to Sprint and set the time window to your last 10 sprints. Export this data to Excel for your Monte Carlo spreadsheet. ADO's built-in chart is useful for trend visualisation but does not natively run Monte Carlo simulations.

5. Monte Carlo in Azure DevOps Three options: - **ActionableAgile for Azure DevOps** (actionableagile.com/analytics): the same tool recommended for Jira, with an ADO connector. Directly the most capable option. - **Nave** (nave.app): connects to ADO and provides flow analytics including Monte Carlo. - **ADO + Excel template**: export throughput data from ADO Analytics, paste into a Monte Carlo spreadsheet template. Manual but free and sufficient for teams not yet at sprint 8.

6. LGP pre-read and Node Health Card Use Azure DevOps Wiki for Node Health Cards and LGP pre-reads. Create a wiki page per ACN, with a standard template covering throughput trend, DOI Health Score, IIS status, and open dependencies. Link the wiki page from the team's ADO dashboard. The decision log can be maintained as a separate wiki page updated by the Governance Steward after each LGP session.

Q: We use Linear. Does SHIFT work with it?

A: Yes. Linear is well-suited to SHIFT teams of 3 to 7 people. It is cleaner to configure than Jira and has good built-in cycle (sprint) support.

SEMI scoring: Add four custom properties to your issue type — S, E, M, I as number fields (1-3), plus a computed SEMI Score display. Linear supports custom properties natively. Add these to your issue template so they are prompted at creation.

DOI Status: Add a custom select property with options Green, Amber, Red. Use Linear's filter and grouping to view DOI status distribution across active cycle issues. This replaces the Jira dashboard gadget.

Throughput tracking: Linear's built-in cycle analytics show issue completion counts per cycle. Use this as your throughput data source. Export to a spreadsheet for Monte Carlo inputs.

IIS work: Use a dedicated Label ("IIS") or a separate Linear project for IIS work within each cycle. Filter IIS issues out of your delivery throughput baseline.

Monte Carlo: Linear does not have a native Monte Carlo tool. Export throughput data and use an ActionableAgile account (which supports CSV import) or a spreadsheet template.

Limitation: Linear's reporting is less configurable than Jira or ADO for large teams. For teams above 7 to 8 people or for multi-node coordination, Jira or ADO is the better choice.

Q: What about Notion, Trello, or spreadsheet-only setups?

A: SHIFT can run on any tooling, including a shared spreadsheet, if the team is disciplined. The minimum viable toolset is:

- A shared spreadsheet with two tabs: (1) Sprint throughput log, one row per sprint with items completed; (2) SEMI backlog, listing each item with S, E, M, I scores and composite.
- A shared document (Google Docs, Notion, or Confluence page) per sprint for the Sprint DOI Map (backlog items with RAG tags, updated mid-sprint).
- A shared document per ACN for the Node Health Card (updated each sprint end).
- A shared document per IIS sprint for Learning Cards.

Trello: Usable for 3 to 5 person teams. Add SEMI Score as a custom field on cards, DOI Status as a label, IIS as a separate list or board. Throughput tracking requires manual counting or a Power-Up. Not recommended as a primary tool above 5 people.

Notion: Good for documentation artefacts (Node Health Cards, Learning Cards, Pattern Library, AI Responsibility Map, LGP pre-reads and decision log). Weak for sprint tracking and throughput analytics. Use Notion for documentation alongside a dedicated board tool for sprint management.

The honest answer: The tool matters far less than the discipline of updating it. A team that maintains a clean shared spreadsheet will get more value from SHIFT than a team that configures a full Jira SHIFT setup and never updates the DOI Status field.

Team and Roles

Q: We do not have a dedicated Governance Steward. Can the Delivery Lead cover both roles?

A: In teams of 5 or fewer people, yes, with caution. The Delivery Lead can carry the Governance Steward function provided two conditions are met: (1) the LGP pre-read is compiled and distributed 24 hours in advance without exception, and (2) the decision log is maintained and visible to all ACNs. If either condition is not being met consistently, the Governance Steward function is being neglected and should be given to someone else, even informally.

The risk of combining these roles: the Delivery Lead's primary loyalty is to sprint delivery. When the two roles conflict, governance gets deprioritised. The most common failure mode is the LGP pre-read not being sent until the morning of the session, which means participants arrive without context and the session becomes a status report rather than a decision forum.

In teams above 5 people, do not combine these roles. The authority concentration creates a single point of failure for both operations and governance.

Q: Can we rotate the Anchor role between team members?

A: Rotating the Delivery Lead role is encouraged (see Chapter 5). Rotating the Anchor role is not recommended and is explicitly discouraged for ACNs in active delivery.

The Anchor role requires deep familiarity with the capability domain's history, the decisions made, the trade-offs accepted, and the strategic direction. This context cannot be effectively transferred in a single handover session. An ACN that rotates its Anchor every cycle will lose continuity precisely when continuity matters most, at cycle boundaries when strategic decisions are made.

The appropriate version of role development is: a future Anchor candidate shadows the current Anchor for a full cycle, participates in LGP sessions, and takes on specific Anchor accountabilities before formally rotating in. This is a development pathway, not a rotation schedule.

Q: What do we do when the Anchor and Delivery Lead persistently disagree on priorities?

A: This is a governance issue, not a personality issue, and it should be resolved at the LGP, not informally.

The role separation is intentional: the Anchor owns outcome direction, the Delivery Lead owns operational execution. When they disagree persistently, it usually means either the capability domain is not clearly enough defined (Anchor and Delivery Lead are working from different mental models of what the ACN owns) or the sprint capacity and commitment is being managed without shared understanding (Delivery Lead is protecting delivery bandwidth that the Anchor is treating as available for scope increases).

The resolution path: bring the specific disagreement to the LGP as an Unblock item. The Governance Steward facilitates a decision that documents the explicit scope and capacity boundary. Once the boundary is written down and ratified, the disagreement usually resolves because both parties are now working from the same explicit constraint rather than implicit assumptions.

Integration and Adoption

Q: Our company runs SAFe at the programme level. Can one team adopt SHIFT without the whole organisation changing?

A: Yes, and this is one of the most common adoption patterns. A single ACN can operate SHIFT internally while participating in SAFe ceremonies externally. The integration works as follows:

- The Anchor participates in SAFe's PO Sync and ART events as the team's representative. They translate SAFe Programme Increment objectives into SHIFT Cycle Objectives and maintain the DOI map internally.
- SAFe's PI Planning outputs become the inputs to SHIFT's first Cycle DOI Map. PI objectives map to Cycle Objectives.
- The team uses SEMI scoring instead of story point estimation internally. For PI Planning outputs and external reporting, the team maintains a story point field that is derived from historical throughput averages, not from bottom-up estimation.
- The LGP operates entirely internally. SAFe's ART Sync serves as the external governance input; the LGP handles internal cross-node decisions.

The benefit of this pattern: the team gets the full SHIFT operating model while remaining a participant in the SAFe structure. After two or three cycles of demonstrably better throughput predictability and alignment quality, the evidence base for expanding SHIFT to other teams or replacing SAFe ceremonies with LGP is significantly stronger.

The risk: SAFe PI Planning events will not be SEMI-scored, which means items entering the team's work from PI Planning may arrive with poor specification quality. Introduce a Spec Review session immediately after PI Planning to score incoming PI objectives before they enter the sprint backlog.

Q: We already have OKRs. Where does the DOI map overlap with what we already track?

A: DOI is not a replacement for OKRs. It is the connection between your OKR system and your sprint work. The typical problem: teams write OKRs, run sprints, and discover at quarter end that there is no traceable link between what was built and what the OKRs intended.

The mapping is direct: - Your company OKRs become inputs to the Alignment layer at cycle start. - Your team OKR (or the relevant Key Result your team owns) becomes the Cycle Objective in the DOI map. - Sprint Contributions in the DOI map are the specific deliverables each sprint is committing to in service of that Key Result. - The DOI Health Score gives you a mid-cycle, sprint-by-sprint view of whether the team's work is actually contributing to the Key Result as planned.

What SHIFT adds that OKRs alone do not provide: a continuous, sprint-level signal of whether execution matches intent, surfaced during the sprint when it can still be corrected, not at quarter end when it cannot.

If your team is already writing OKRs, the transition to DOI is low effort: the Cycle Objective is your OKR at team level, and the Sprint Contributions are the specific commitments that connect sprint delivery to it. The only new practice is the mid-sprint DOI check, where those contributions are tagged Green, Amber, or Red based on current progress.

Q: How do we handle stakeholders who want predictable release dates but do not understand Monte Carlo?

A: Most stakeholders do not need to understand Monte Carlo. They need a confident, honest answer to "when will it be done?"

The communication formula: "Based on our delivery data from the last [N] sprints, we have 85% confidence this will be complete by [date]. Our most likely completion, based on current throughput, is [earlier date]."

This gives stakeholders two numbers: a planning date (P85, the one to put in the roadmap or commit to externally) and an optimistic date (P50 or P70, the one the team is actually working toward). Most stakeholders accept this framing without needing to understand the underlying simulation.

What to avoid: giving a single point estimate ("we'll be done by March 14") with false precision, or giving a wide range without a confidence statement ("somewhere between February and April") that sounds evasive. The Monte Carlo language gives precision with honesty: a specific date with an explicit probability.

For stakeholders who push back on probabilistic language: reframe it in terms they already use. "It is the same logic as a weather forecast. We are not saying it will definitely be done by March 14. We are saying that based on our actual delivery patterns, 85 times out of 100

it would be done by then. We will give you a heads-up the moment we see signals that put that date at risk."

Q: What is the most common reason SHIFT implementations fail in the first cycle?

A: There are three failure modes that account for the majority of first-cycle problems:

Failure mode 1: Introducing too much at once. Teams attempt to run IIS, DOI, LGP, Monte Carlo, SEMI, the AI-first standup, and the 3-track retrospective in sprint 1. The cognitive and process overhead is too high. Sprint 1 becomes a meta-conversation about SHIFT instead of a delivery sprint. The fix is the phased adoption plan in Chapter 21. Strictly follow it.

Failure mode 2: SEMI scoring without honest disagreement. Teams go through the motions of scoring SEMI but socially align on scores rather than surfacing real disagreement. When two people give the same item S=1 and S=3, the right response is to explore why, not to average to S=2. The value of SEMI is in the forced clarification conversation. If SEMI scores are not causing any disagreement, either the team has unusually clear specs or the scoring is being done dishonestly.

Failure mode 3: LGP pre-read not enforced. The LGP session is called, the pre-read is not ready, the session goes ahead anyway as a verbal status update, and the pattern sets in. Once LGP becomes a status report forum, it is very difficult to recover without an explicit reset. The Governance Steward must enforce the pre-read non-negotiable from the very first session. If the pre-read is not ready, the session is postponed, not held without it.

Closing Note

SHIFT is a living framework. It was designed to evolve through use. The pattern library concept at Levels 3 and 4 allows teams to feed their practical experience back into the framework's calibration tables, ceremony formats, and SEMI scoring guides.

The framework's core commitments do not change across integrations, team sizes, or maturity levels:

Delivery and exploration are not trade-offs. IIS proves this.

Governance should cost less than it prevents. LGP is designed to that constraint.

Strategy and execution must be continuously connected, not periodically reconciled. DOI is the mechanism.

AI is a team member with defined responsibilities, not a tool with undefined outputs. The AI Responsibility Map codifies this.