#	Framework	1. Agile Category	2. Core Intent	3. Primary Focus	4. Roles (Accountabilities)	5. Cadence	6. Core Practices / Ceremonies	7. Key Artifacts	8. Work Management	9. Primary Metrics	10. Org Fit	11. Strengths	12. Failure Modes	13. Best Used When...
1	**Scrum**	Team-level Framework	Iterative delivery of complex products.	Value Delivery via Empirical Process Control.	PO, Scrum Master, Developers.	Fixed Sprints (1–4 weeks).	Sprint Planning, Daily Scrum, Review, Retrospective.	Product Backlog, Sprint Backlog, Increment.	Iterative Pull.	Velocity, Sprint Goal Success.	Small cross-functional teams; complex product environments.	High transparency; clear accountability; rapid risk mitigation.	"Zombie Scrum" (ritual without mindset); "Scrum-but"; Lack of psychological safety.	You need to validate product ideas quickly with a dedicated team in a complex environment.
2	**XP**	Engineering Centric	High-quality code via technical excellence.	Technical Discipline & Reducing Cost of Change.	Customer, Programmer, Coach, Tracker.	Weekly Cycles.	Pair Programming, TDD, CI/CD, Refactoring.	User Stories, Automated Tests, Code.	Iterative Pull (Planning Game).	Code Quality, Test Coverage, Velocity.	Small co-located teams building software with high quality needs.	Lowest defect rates; sustainable pace; high adaptability.	Cultural resistance to pairing; skipping tests under pressure.	Technical risk is high and the cost of software failure is unacceptable.
3	**Kanban**	Lean / Flow	Optimize flow and reduce waste.	Flow Efficiency & Reducing Lead Time.	No prescribed roles (Evolutionary).	Continuous Flow.	Visualize, Limit WIP, Manage Flow, Feedback Loops.	Kanban Board, WIP Limits, CFD.	Continuous Pull.	Cycle Time, Throughput, WIP, SLE.	Service/Support teams, or mature product teams optimizing flow.	Visualizes bottlenecks; handles variability well; low resistance to start.	Over-commitment (ignoring WIP limits); lack of improvement focus.	Work is unpredictable (support/ops) or you want to optimize an existing process without a re-org.
4	**FDD**	Process Oriented	Scaling through feature-centric modeling.	Domain Modeling & Feature Ownership.	Chief Architect, Chief Programmer, Class Owners.	Feature-based (2-14 days).	Domain Modeling, Develop by Feature, Inspections.	Features List, Object Model, Design Packages.	Plan by Feature (Push/Pull hybrid).	Feature Progress (%), Build Health.	Large teams; complex business domains; bank/financial systems.	Scalability; tracking; architectural integrity.	Siloing of class owners; heavy upfront modeling time.	You have a very large team building a complex object-oriented system requiring strict architectural control.
5	**DSDM**	Governance	Discipline and on-time delivery.	Strategic Alignment & Fixed Constraints.	Sponsor, Visionary, PM, Technical Coordinator.	Fixed Timeboxes.	MoSCoW Prioritization, Timeboxing, Workshops.	Business Case, Prioritized Req. List (PRL), Architecture.	Fixed Time/Cost, Variable Scope.	Value Realization, On-time Delivery.	Corporate/Government with strict fixed deadlines/budgets.	Strong governance; predictability; business engagement.	Too much bureaucracy; treating it as Waterfall.	You are in a corporate/government environment requiring fixed deadlines and auditable governance.
6	**Crystal**	Methodology Family	Human-centric, lightweight delivery.	Communication & Team Safety.	Variable (based on color/team size).	Frequent Delivery.	Osmotic Communication, Reflective Improvement.	Project Map, Release Plan, Status Reports.	Team-defined.	Delivery Frequency, Morale.	Co-located teams; varies by size (Clear, Yellow, Orange).	Extremely lightweight; high morale; adaptable.	Lack of discipline; failure to reflect and improve.	You need a lightweight, custom approach for a small, co-located team (Crystal Clear).
7	**AUP**	Hybrid / Legacy	Bridging RUP and Agile.	Lifecycle Coverage (Serial-Large, Iterative-Small).	Modelers, Implementers, Project Manager.	Phased (Inception to Transition).	Model Storming, Test in the Small.	Agile Models, Source Code, Test Suite.	Phase-based milestones.	Phase Milestones, Defect Trends.	Organizations transitioning from heavy RUP/Waterfall.	Familiar structure for traditional managers; discipline.	Heavy documentation; "Water-Scrum-Fall".	You are transitioning a legacy RUP organization or have heavy regulatory documentation needs.
8	**Scrumban**	Hybrid	Transitioning from Scrum to Flow.	Capacity Planning & Flexibility.	Retains Scrum roles (mostly).	Continuous or Sprints.	On-demand Planning, WIP Limits, Daily Sync.	Scrumban Board, Backlog (Trigger-based).	Pull-based with planning triggers.	Cycle Time, Lead Time.	Maintenance teams; product teams tired of Sprints.	Flexibility of Kanban with structure of Scrum.	Loss of long-term planning; "lazy" Scrum.	Your Scrum team feels constrained by Sprints, or you handle a mix of roadmap and support work.
9	**Shape Up**	Product-Led Value Stream Orchestration	Break the "feature factory" treadmill.	Shaping (Defining) & Betting (Risk Mgmt).	Shapers, Builders (Designers/Devs).	6-Week Cycle + 2-Week Cool-down.	Shaping, Betting Table, Hill Charts, The Circuit Breaker.	The Pitch, Hill Charts, Fat Marker Sketches.	Betting (No Backlog). Fixed Time, Variable Scope.	Hill Chart Progress, Shipped Bets.	Product companies; startups; mature dev teams.	Focus; no backlog grooming; meaningful work slices.	Shapers disconnecting from reality; "mini-waterfall".	You want to stop backlog grooming and focus on finishing meaningful projects in fixed timeboxes.
10	**SoS**	Scaling (Basic)	Inter-team coordination.	Synchronization of dependencies.	Ambassador, SoS Master.	Daily or Weekly sync.	The SoS Meeting, Cross-team Retrospective.	Impediment Backlog.	Coordinated Team Pull.	Dependency Resolution Rate.	Multiple teams (3-9) on one product.	Lightweight; familiar; low cost.	Becomes a status report; ignores systemic issues.	You have a few teams (3-9) working on a single product and need a lightweight way to connect.
11	**SAFe**	Scaling (Enterprise)	Synchronizing alignment at scale.	Portfolio Alignment & Predictability.	RTE, Product Mgmt, System Arch, SPC.	10-week Planning Interval (PI).	PI Planning, System Demo, Inspect & Adapt.	ART Backlog, Program Board, Arch. Runway.	Program-level Pull (PI Objectives).	Flow Predictability, Flow Load.	Global 1000; highly regulated or complex systems.	Alignment; funding/strategy connection; common language.	"SAFe-washing" (bureaucracy); heavy overhead; slow decisions.	You are a Global 1000 enterprise with hundreds of devs needing strict alignment between strategy and execution.
12	**LeSS**	Scaling (Descaling)	Scrum applied to multiple teams.	Whole Product Focus & Simplification.	1 PO, Scrum Master, Feature Teams.	Synchronized Sprints.	Joint Sprint Planning, Overall Retro, Joint Review.	Single Product Backlog, Integrated Increment.	Team Pull from shared backlog.	Value Delivery, Cycle Time.	Product-centric orgs willing to restructure.	Simplicity; customer focus; minimal overhead.	Political resistance to removing management layers.	You want to scale agile by *removing* bureaucracy and organizational complexity (Descaling).
13	**DA**	Decision Toolkit	Context-driven choice (Choose Your WoW).	Process Tailoring & Flexibility.	Team Lead, Architecture Owner.	Variable (Lifecycle dependent).	Guided Continuous Improvement, Goal Selection.	Way of Working (WoW) agreement.	Context-dependent.	G6 Metrics (Value, Quality, etc.).	Organizations with diverse team needs.	Adaptability; agnostic; enterprise awareness.	Analysis paralysis (too many choices); lack of consistency.	You have diverse teams with different needs and want a toolkit to help them optimize their own way of working.
14	**Team Topologies**	Org Design	Optimize team interactions for flow.	Cognitive Load & Fast Flow.	Stream-aligned, Enabling, Platform, Subsystem.	Evolutionary (Triggered by flow).	Interaction Modes (Collaboration, XaaS, Facilitating).	Team API, Thin Platform.	Flow-based; Interaction-driven.	Team Cognitive Load, Flow Efficiency.	Modern software orgs; DevOps/Platform engineering.	Reduces burnout; clear boundaries; enables fast flow.	Creating silos; renaming old teams without changing behavior.	Your teams are burnt out from cognitive overload or blocked by endless hand-offs.
15	**Flight Levels**	Strategy Model	Connect strategy to execution.	End-to-End Flow across the org.	No prescribed roles (Uses existing).	Nested Cadences (L1, L2, L3).	Visualizing Topology, Strategy Boards, Coordination.	Flight Level Boards (L1, L2, L3).	Flow connections between levels.	Time-to-Market, Strategy Execution.	Any org with disconnected strategy and execution.	No re-org needed; connects silos; highlights bottlenecks.	Only visualizing L1 (teams); ignoring L3 (strategy).	You have "Agile Teams" (Level 1) but no "Business Agility" because strategy and coordination are disconnected.

Note: Lean Thinking is excluded from this matrix as it is the foundational philosophy for all methods. Similarly, the **unFIX** Model and **Agentic AI** Governance are excluded because they represent emerging structures and future capabilities rather than established delivery frameworks.