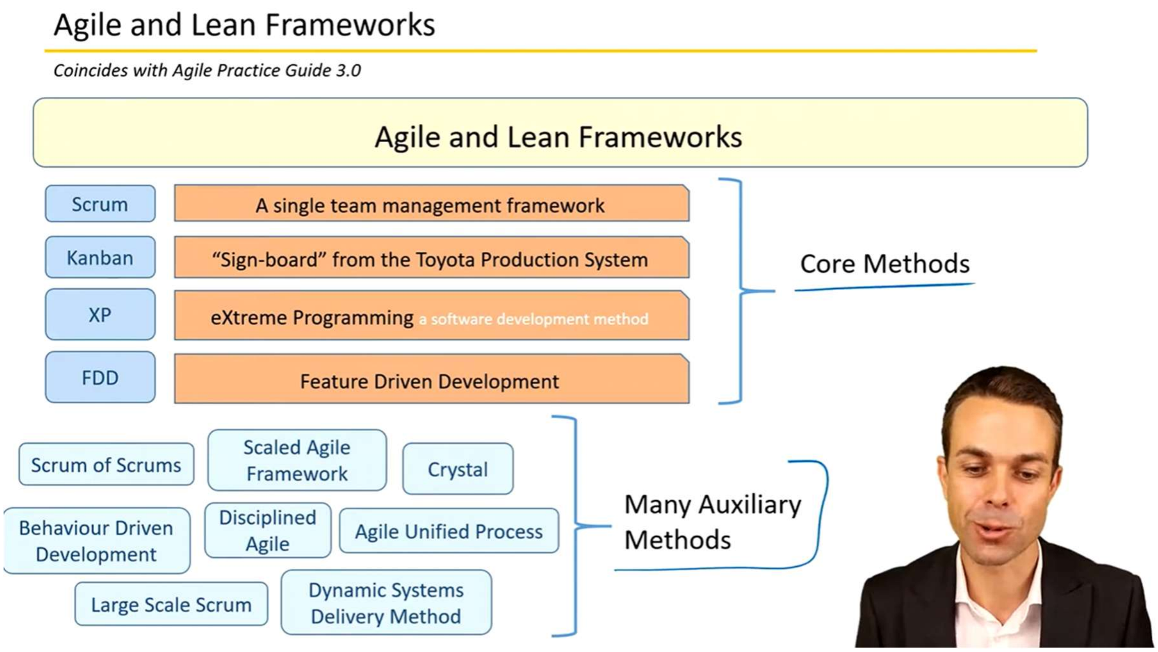
Agile and Lean Frameworks

Readable Study Guide

This guide summarizes Agile core methods and auxiliary methods in a structured, exam-friendly format.

Table of Contents



Agile and Lean Frameworks

Coincides with Agile Practice Guide 3.0

Agile and Lean Frameworks

Scrum — A single team management framework

Kanban — "Sign-board" from the Toyota Production System

XP — eXtreme Programming a software development method

FDD — Feature Driven Development

Core Methods

Scrum of Scrums

Scaled Agile Framework

Crystal

Behaviour Driven Development

Disciplined Agile

Agile Unified Process

Large Scale Scrum

Dynamic Systems Delivery Method

Many Auxiliary Methods

1. Agile Core Methods

1.1 Scrum

Scrum is a single-team management framework for managing product development.

The Scrum Team typically includes three key roles:

- Product Owner – represents the customer, grows and prioritizes user stories on the backlog.
- Development Team – develops, tests, and delivers the product increment.
- Scrum Master – ensures Scrum practices are followed, facilitates events, and removes blockers.

Scrum Events

#	Event	Definition
1	Sprint	A time-boxed project iteration, usually two to four weeks.
2	Sprint Planning	At the start of each sprint, the Scrum Team selects the highest-priority items from the Product Backlog.
3	Daily Scrum	A short (often 15-minute) stand-up meeting to walk through tasks and progress, often using a Kanban board.
4	Sprint Review	The Development Team demonstrates the increment to the Product Owner and stakeholders for feedback, sign-off, or rework.
5	Sprint Retrospective	A meeting at the end of the sprint to reflect and improve the way of working for the next iteration.

Scrum Artifacts

#	Artifact	Definition
1	Product Backlog	A prioritized list of planned product items, owned and managed by the Product Owner, which evolves over time.
2	Sprint Backlog	The set of Product Backlog items selected for the upcoming sprint during Sprint Planning.
3	Increment	All Product Backlog items completed during a sprint, representing a step toward the product goal or main release.

1.2 Kanban

Kanban (Japanese for 'sign-board') originates from the Toyota Production System.

It is a form of visual management from Lean manufacturing, used to monitor work in progress and enable pull and flow.

- Pull – team members pull new work only when they are ready, rather than allowing inventory or unfinished work to pile up.
- Flow – work moves smoothly through the value stream with minimal waiting and rework.
- Kanban is particularly helpful when you need:
- Increased efficiency – visibility of each task and ensuring that it adds value.
- Better team focus – limiting work in progress (WIP) to avoid multitasking and context switching.
- Handling variability in workload – visualizing and smoothing incoming work.
- Reduction of waste – transparency allows the team to spot and remove non-value-adding activities.

The Kanban board acts as an information radiator, providing up-to-date status of work items to anyone who views it.

Kanban Principles and Core Properties

#	Defining Principle	Core Properties (Examples)
1	Start with the current state	Visualize the workflow.
2	Agree to pursue incremental, evolutionary change	Limit work in progress (WIP).
3	Respect the current process	Manage flow across the system.
4	Lead at all levels	Enable pull and empower people to improve.
5		Make process policies explicit.
6		Implement feedback loops.
7		Improve collaboratively and evolve experimentally.

1.3 eXtreme Programming (XP)

XP is a software development method based on frequent cycles and strong engineering discipline.

It popularized a holistic set of core practices and emphasizes lightweight, human-centered, disciplined development.

XP stresses frequent communication with the customer, continuous testing, and regular refactoring to improve design.

XP Practice Areas

Practice Area	Primary Practices	Secondary Practices
Organizational	Sit together; Whole Team; Informative workspace	Real customer involvement; Team continuity; Sustainable pace
Technical	Pair programming; Test-first programming; Incremental design	Shared code / collective ownership; Documentation from code and tests; Refactoring
Planning	User stories; Weekly cycle; Quarterly cycle; Slack	Root cause analysis; Shrinking teams; Pay per use; Negotiated scope contract; Daily stand-ups
Integration	10-minute build; Continuous integration; Test-first	Single code base; Incremental deployment; Daily deployment

XP is often compared with Scrum and Kanban: Scrum focuses on managing the process and iterations, Kanban on visual flow and WIP, and XP on disciplined engineering practices.

1.4 Feature Driven Development (FDD)

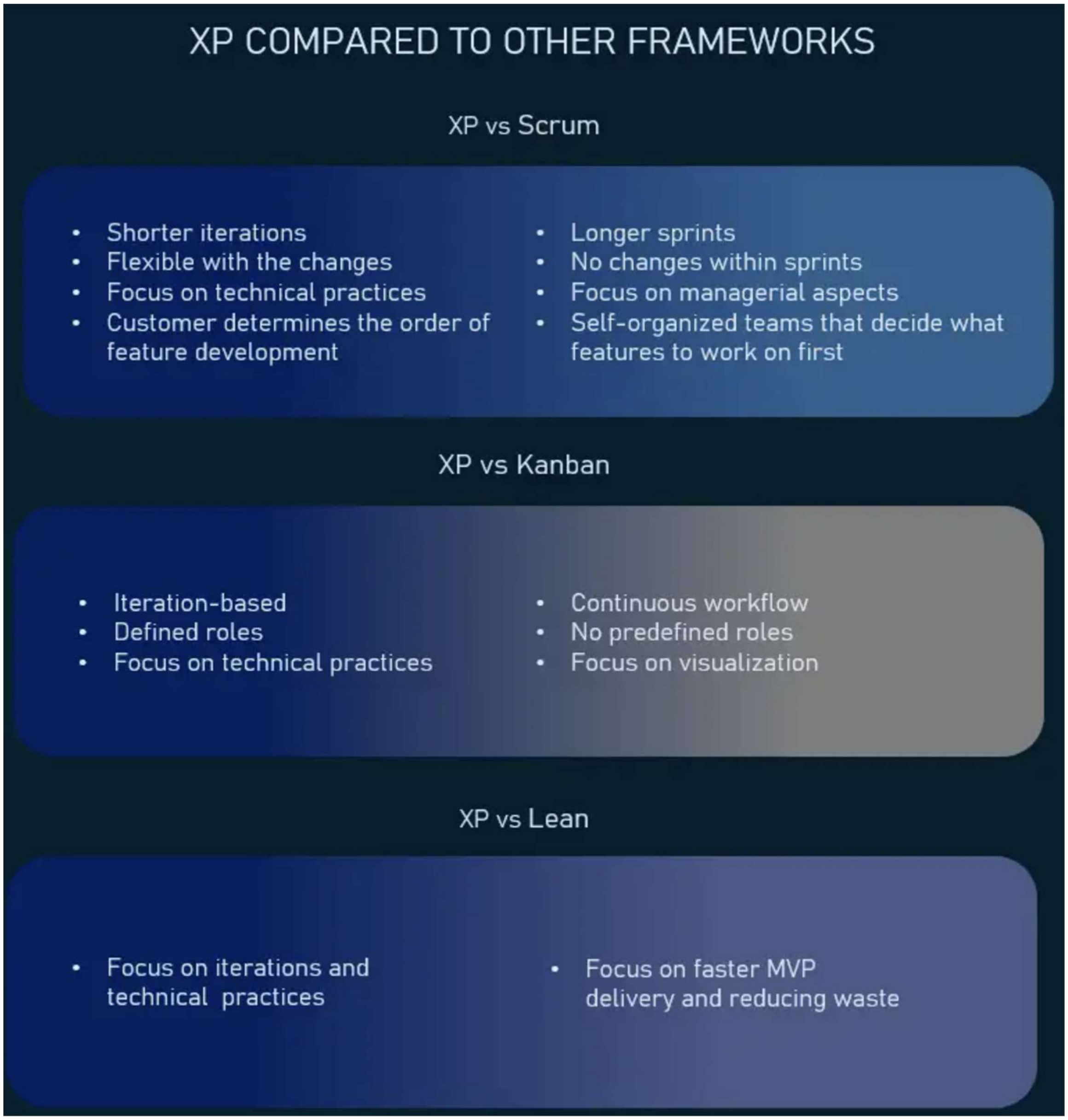
Feature Driven Development (FDD) is both iterative and incremental. It emphasizes planning and building around features.

- Iterative – feedback on features is rolled back into the product to improve over time.
- Incremental – features are developed and delivered so customers can see and touch working functionality.

Typical FDD activities include:

- Developing an overall (high-level) model.
- Building a comprehensive features list.
- Planning by feature.
- Designing by feature (e.g., using storyboards).
- Building by feature and incorporating customer feedback.

Refer to the original README for the FDD flow diagram (Develop model → Feature list → Plan → Design → Build).



XP COMPARED TO OTHER FRAMEWORKS

XP vs Scrum

- Shorter iterations
- Flexible with the changes
- Focus on technical practices
- Customer determines the order of feature development

- Longer sprints
- No changes within sprints
- Focus on managerial aspects
- Self-organized teams that decide what features to work on first

XP vs Kanban

- Iteration-based
- Defined roles
- Focus on technical practices

- Continuous workflow
- No predefined roles
- Focus on visualization

XP vs Lean

- Focus on iterations and technical practices

- Focus on faster MVP delivery and reducing waste

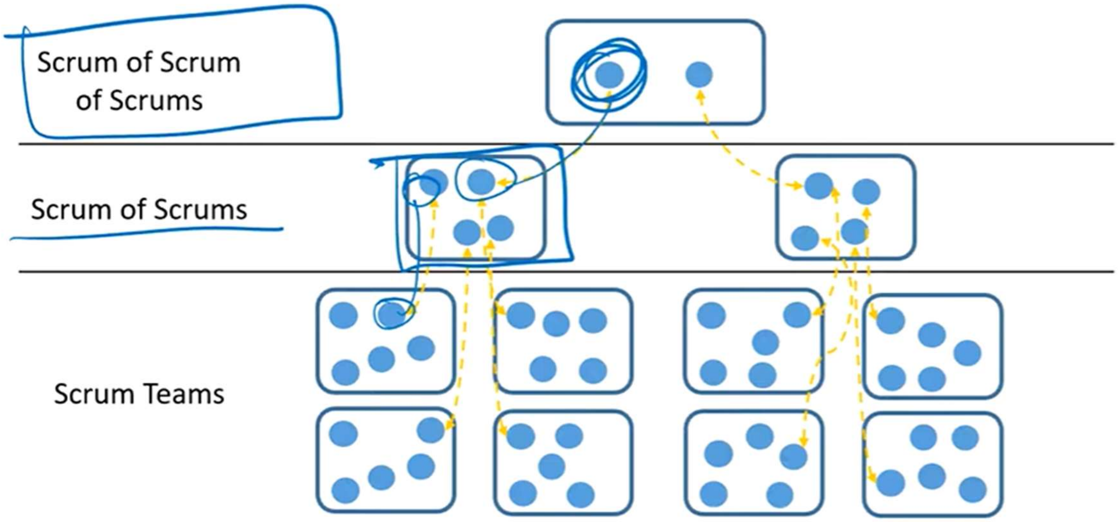
2. Agile Auxiliary Methods

2.1 Scrum of Scrums

- Scrum of Scrums is used when multiple Scrum teams (typically 3–9 people each) must coordinate their work.
- A representative from each team attends a Scrum of Scrums meeting several times per week.
- In these meetings, representatives typically discuss:
- Completed work since the last Scrum of Scrums.
- Upcoming work for their team.
- Current blockers.
- Potential upcoming blockers that may affect other teams.

The goal is to ensure cross-team coordination and removal of cross-team impediments. It aligns with the Projects → Programs → Portfolios view in the PMBOK Guide.

See the original diagram in the README for a visual representation of Scrum of Scrums.



2.2 Scaled Agile Framework (SAFe)

SAFe is a framework for applying Agile at scale across an enterprise.

It focuses on organizing around value streams at the portfolio, program, and team levels.

- Key principles include:
- Take an economic view when making decisions.
- Apply systems thinking – understand how small pieces fit into the overall system.
- Assume variability and preserve options.
- Build incrementally with fast, integrated learning cycles.
- Base milestones on objective evaluation of working systems.
- Visualize and limit work in progress, reduce batch sizes, and manage queue lengths (often using Kanban boards).
- Apply cadence and synchronize with cross-domain planning (e.g., Scrum of Scrums, PI planning).
- Unlock the intrinsic motivation of knowledge workers.
- Decentralize decision making where appropriate

2.3 Crystal

Crystal is a family of Agile methods focusing on people, interactions, and communication rather than rigid processes.

Introduced by Alistair Cockburn, it emphasizes tailoring practices based on team size and project criticality.

- Three core beliefs:
- Technologies change techniques.
- Cultures change norms.
- Distances change communication (face-to-face is preferred when possible).

Crystal Values and Common Properties

Core Values / Focus	Common Properties
People and interaction	Frequent delivery; close communication; focus; personal safety.
Community and skills	Reflective improvement (retrospectives); easy access to expert users (e.g., via Product Owner).
Technical environment	Automated tests; configuration management; frequent integration.

Crystal is designed to scale; different 'colors' (e.g., Clear, Yellow, Orange, Red) reflect different team sizes and project criticality.

2.4 Behavior Driven Development (BDD)

BDD defines stories and features from the customer’s point of view.

It involves the whole team in specifying behavior through examples and tests.

BDD frameworks often use the Given / When / Then format to define acceptance criteria.

- Typical structure:
- Given some initial context,
- When an event or action occurs,
- Then verify expected outcomes.

In diagrams, this is often shown as: Given → When → Then.

2.5 Disciplined Agile (DA)

Disciplined Agile is a process decision framework that blends practices from multiple Agile and Lean methods.

DA Principles

Principle	Description
People First	Enumerates roles and organizational elements at various levels; focuses on individuals and interactions.
Learning-Oriented	Encourages collaborative improvement and continuous learning.
Full Delivery Life Cycle	Promotes several fit-for-purpose life cycles across the organization.
Goal-Driven	Tailors processes to achieve specific outcomes rather than following a single fixed method.
Enterprise Awareness	Offers guidance on cross-departmental governance and alignment.
Scalable	Covers multiple dimensions of program complexity and organizational scale.

2.6 Agile Unified Process (AUP)

AUP adapts the Unified Process in a more Agile, lightweight way. It emphasizes iterative cycles across seven key disciplines and incorporates feedback before formal delivery.

Discipline	**Guiding Principle**
Model	The team understands what it is building.
Implementation	Simplicity in design and construction.
Test	Emphasis on agility and rapid feedback.
Deployment	Focus on high-value activities and prioritizing the backlog.
Configuration Management	Tool independence and effective version control.
Project Management	Tailoring the approach to fit the situation.
Environment	Situationally specific tooling and infrastructure.

2.7 Large Scale Scrum (LeSS)

LeSS extends Scrum for use with multiple teams working on the same product. It keeps Scrum simple while adding just enough guidance for scaling.

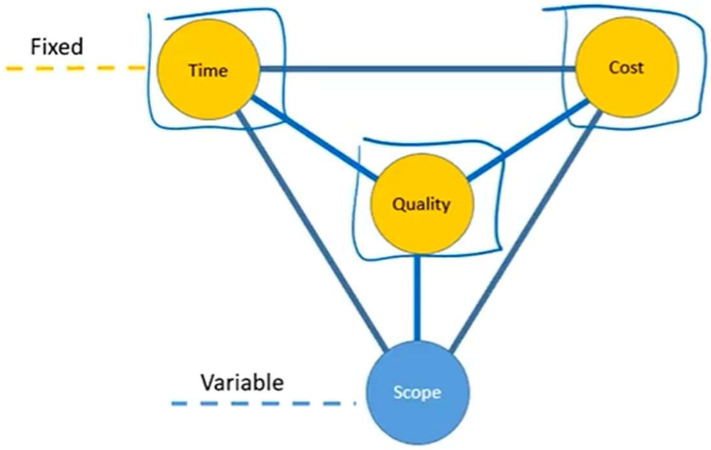
Scrum-like Elements in LeSS	Additional LeSS Techniques
One single Product Backlog	Overall cross-team refinement to coordinate backlog items among teams.
One Definition of Done for all teams	Overall retrospectives focused on cross-team improvements.
One potentially shippable increment per sprint	More formal division of sprint planning into 'what' and 'how' across teams.
One Product Owner	Organic cross-team coordination and communication patterns.
Complete, cross-functional teams	
One sprint for all teams	

2.8 Dynamic Systems Delivery Method (DSDM)

DSDM adds more rigor to iterative methods and is known for its constraint-driven delivery.

Cost, quality, and time are fixed; scope is flexed using formal prioritization to meet those constraints.

- Eight principles of the DSDM framework:
- Focus on the business need (deliver products frequently, often every 2–4 weeks).
- Deliver on time.
- Collaborate.
- Never compromise quality.
- Build incrementally from firm foundations.
- Develop iteratively to enable customers to see progress.
- Communicate continuously and clearly (e.g., daily stand-ups).
- Demonstrate control through appropriate techniques and visible progress.



2.9 Enterprise Scrum

Enterprise Scrum applies Scrum principles and practices at the organizational level, not just at the single-team product level.

It guides leaders to extend and generalize Scrum techniques across many domains and levels of the organization.

- It advises leaders to:
- Extend the use of Scrum across departments and functions.
- Generalize Scrum techniques so they can be applied at team, program, and portfolio levels.
- Scale Scrum with supplemental techniques (e.g., additional coordination practices) as necessary.