Agile Software Project Management

Instructor – Muhammad Sudais

Additional Agile Methods

Chapter 17

Additional Agile Methods

- Agile **evolved** beyond Scrum and XP to **address diverse project needs**.
- Different methods focus on **speed, efficiency, risk management, or scalability**.
- These alternative Agile methods are widely used in **various industries** beyond software development.
- This chapter explores methodologies like **DSDM, Crystal, Lean, Kanban, and FDD**.

Dynamic Systems Development Method (DSDM)

- Originated from **Rapid Application Development (RAD)** in the **UK (1994)**.
- Provides a **structured Agile framework** for **business-critical projects**.
- **Principles:**
  - Business **needs drive** the project.
  - **User involvement** is essential.
  - **Frequent deliveries** with prioritization.
  - **Iterative and incremental development**.
  - **MoSCoW prioritization** (Must-have, Should-have, Could-have, Won't-have).
  - **Integrated testing** in each development cycle.
- **Best for: Government, financial, and large-scale enterprise projects**.

•**Example:** Used in **banking projects** to ensure compliance and rapid feature development.

Crystal Methods

- **Created by Alistair Cockburn**; focuses on **people, not processes**.
- Designed for projects of **different sizes and criticality**.
- **Color-coded methods:**
  - **Crystal Clear**: Small teams, lightweight processes.
  - **Crystal Yellow, Orange, Red**: Larger teams with increased rigor.
- **Key Focus Areas:**
  - **Frequent delivery** of working software.
  - **Reflective improvement** after each cycle.
  - **Ongoing communication** with minimal documentation.
- **Best for: Teams needing flexibility & scalability**.
- **Example:** Used in **healthcare and mission-critical systems** (e.g., Electronic Medical Records).

Agile Modeling (AM)

- A lightweight **modeling and documentation approach**.
- **Guiding Principles:**
  - Keep documentation **"Just Barely Good Enough" (JBGE)**.
  - **Active stakeholder participation** throughout.
  - **Iterate quickly** to adapt models over time.
  - Support **Test-Driven Development (TDD)**.
- **Best for:** Teams that need **flexible architecture planning** without **heavy documentation**.

•**Example:** Used in **early-stage software architecture planning**.

Lean Development

- Inspired by **Toyota's Lean Manufacturing** methodology.
- **Goals:**
  - **Eliminate waste** in processes.
  - **Deliver value quickly** by focusing on essential features.
  - **Empower the team** to make key decisions.
  - **Continuous learning & improvement**.
- **Best for: Startups, continuous delivery teams, AI/ML projects**.

•**Example:** Used by **Tesla** for rapid feature rollout & minimal production delays.

Adaptive Software Development (ASD)

- Evolution of **Rapid Application Development (RAD)**.
- **Three core principles:**
  - **Speculate** → Plan for **uncertainty** and expect changes.
  - **Collaborate** → Cross-functional teamwork ensures flexibility.
  - **Learn** → Continuous feedback improves outcomes.
- **Best for: AI-driven projects, IoT, and emerging tech applications**.

•**Example:** Used by **Amazon** for **personalized recommendations & A/B testing**.

Kanban

- **Visual-based workflow management** from **Toyota's Just-in-Time (JIT)** system.
- **Key Features:**
  - **Work-in-Progress (WIP) Limits** → Prevents overload.
  - **Continuous Flow** → Tasks move when capacity allows.
  - **Pull System** → No work starts until previous tasks are completed.
- **Best for: IT operations, DevOps, maintenance teams**.

•**Example:** Used in **Netflix's DevOps team** to manage streaming infrastructure updates.

Just-In-Time (JIT) Development

- Produces **only what is needed, when needed, and in the amount needed**.
- **Benefits:**
  - Reduces waste & inventory costs.
  - **Increases efficiency** through precise demand forecasting.
  - **Supports Agile's incremental delivery model**.
- **Best for: Logistics, software release management, and cloud computing**.

•**Example: Amazon Web Services (AWS)** scales infrastructure dynamically based on demand.

Rapid Product Development (RPD)

- **Short iterative cycles for quick market entry**.
- Uses:
  - **Physical Prototyping** → Test usability & manufacturing feasibility.
  - **Digital Prototyping** → Software/AI model validation.
- **Best for: Hardware, automotive, consumer tech, robotics**.

•Example: Apple develops iPhone prototypes rapidly using RPD principles.

Feature-Driven Development (FDD)

- Focuses on **small, client-valued features**.
- **Five-step process:**
  - Build an overall model.
  - Create a feature list.
  - Plan by feature.
  - Design by feature.
  - Build by feature.
- **Best for: Banking, large-scale enterprise applications**.

•**Example: Used by JP Morgan Chase for financial transaction systems**.

Future Implications for Agile Methods

- **Agile Project Management** → More integration with **AI & automation**.
- **Software Development Life Cycles (SDLCs)** → Agile increasingly replacing Waterfall.
- **Predefined vs. Situation-Specific Guidance** → Shift towards **adaptive Agile approaches**.
- **Empirical Support** → Data-driven Agile adoption is growing.

•**Example:** Hybrid approaches like **ScrumBan (Scrum + Kanban)** are rising in popularity.

Chapter Summary

- **DSDM** → Prioritization & time-boxing.
- **Crystal Methods** → Tailored to team size & risk.
- **Agile Modeling (AM)** → Light, flexible documentation.
- **Lean Development** → Minimize waste, maximize value.
- **ASD** → Adaptive & incremental.
- **Kanban** → Visual workflow management.
- **JIT** → Demand-driven production.
- **RPD** → Rapid innovation cycles.
- **FDD** → Feature-focused iterative delivery.