Summary of Lecture 11: Lean Development

This lecture covers **Lean Software Development**, which originates from **Lean Manufacturing (Toyota Production System)**. Lean focuses on **eliminating waste, improving efficiency, and delivering value to customers**. The core idea is that **anything that does not add value to the customer is considered waste and should be removed**.

What is Lean Software Development?

🔹 **Definition**

- Lean **adapts principles from manufacturing to software development**.
- Originally described by **Mary & Tom Poppendieck**.
- Lean development follows **7 key principles** to **maximize value & minimize waste**.

🔹 **Key Idea**:
✅ **"Deliver fast, reduce waste, and continuously improve."**

The 7 Principles of Lean Development

1️⃣ Eliminate Waste ♻️

- Anything that **does not add value** is waste.
- **Types of Waste (Muda):**
  - ✅ **Unnecessary features** (unused functionality).
  - ✅ **Waiting time** (delays due to dependencies).
  - ✅ **Bottlenecks** (workflow congestion).
  - ✅ **Rework** (caused by poor requirements or testing).

📌 **Example:** Removing excessive documentation that nobody reads.

2️⃣ Amplify Learning 📖

- Software development is an **ongoing learning process**.
- **How to amplify learning?**
  - ✅ **Frequent feedback** (prototypes, demos, early testing).
  - ✅ **Pair programming & code reviews**.
  - ✅ **Retrospective sessions** (reflecting on mistakes & improvements).

📌 **Example:** Running tests as soon as code is written to detect issues early.

3️⃣ Decide as Late as Possible 🕒

- Keep **options open** for as long as possible.
- Helps deal with **uncertainty & changing requirements**.
- **Key techniques:**
  - ✅ **Iterative approach** – Flexibility in decision-making.
  - ✅ **Set-based design** – Experiment with multiple ideas before committing.

📌 **Example:** Choosing a database **only when** the system scales up, instead of prematurely committing to one.

4️⃣ Deliver as Fast as Possible 🚀

- The **faster** software is delivered, the **sooner customers give feedback**.
- Encourages **continuous deployment & quick iterations**.
- **How to achieve fast delivery?**
  - ✅ **Short iterations (Agile, Scrum, Kanban)**.
  - ✅ **Just-In-Time (JIT) development** – Work only on what is needed **right now**.

📌 **Example:** Delivering a **Minimum Viable Product (MVP)** quickly to get real user feedback.

5️⃣ Empower the Team 💪

- Traditional management relies on **top-down control**.
- Lean promotes **self-organizing teams** where **developers make decisions**.
- **How to empower teams?**
  - ✅ **Managers remove obstacles, not micromanage**.
  - ✅ **Developers have direct access to customers for clarification**.
  - ✅ **Flat hierarchy where individuals take responsibility**.

📌 **Example:** Developers **prioritize their own tasks** instead of waiting for management approval.

6️⃣ Build Integrity In 🔧

- Software should be **robust, maintainable, and scalable**.
- **Two types of integrity:**
  - ✅ **Perceived integrity** – The system **feels reliable** to users.

✅ **Conceptual integrity** – The system is **well-structured & works efficiently**.

- **Key practices:**
  - ✅ **Test-Driven Development (TDD)** – Ensures correctness.
  - ✅ **Continuous Integration (CI)** – Reduces system decay.
  - ✅ **Refactoring** – Keeps code clean & adaptable.

📌 **Example:** Writing automated tests **before coding** to ensure software reliability.

7️⃣ See the Whole 🌍

- Software is **not just individual components** but **a system**.
- Teams must **understand the big picture** instead of focusing only on their small parts.
- **How to see the whole?**
  - ✅ **Cross-team communication** – Ensure smooth integration.
  - ✅ **System-wide testing** – Check interactions between modules.
  - ✅ **Avoiding short-term optimizations that harm the long-term system**.

📌 **Example:** A feature might work well **in isolation** but fail when integrated into the full system.

The 7 Wastes in Software Development (Muda)

1️⃣ **Transport** – Unnecessary movement of resources.
2️⃣ **Inventory** – Unused code, unused backlog items.
3️⃣ **Motion** – Unneeded switching between tasks.
4️⃣ **Waiting** – Delays due to approvals, dependencies.
5️⃣ **Overproduction** – Building features that aren't needed.
6️⃣ **Overprocessing** – Rewriting the same code unnecessarily.
7️⃣ **Defects** – Bugs that require fixing later.

📌 **Example:** Writing excessive documentation that **nobody reads or uses**.

Lean & Agile: How They Connect

Lean Principle	Agile Practice
Eliminate Waste	Backlog prioritization (Scrum), removing unnecessary tasks
Amplify Learning	Continuous feedback, retrospectives
Decide Late	Iterative development, late binding

Deliver Fast	Short Sprints, Kanban
Empower the Team	Self-organizing teams
Build Integrity In	Test-Driven Development, CI/CD
See the Whole	System-wide testing, DevOps

✅ **Lean is compatible with Agile but focuses more on efficiency & waste reduction.**

Final Takeaways

✅ **Lean focuses on removing waste, speeding up development, and empowering teams.**
✅ **Software should be delivered quickly and improved continuously.**
✅ **Decisions should be delayed until all necessary information is available.**
✅ **Self-organizing teams perform better than top-down managed teams.**
✅ **A holistic view of the system ensures long-term success.**

Keywords

- **Lean Software Development**
- **Toyota Production System (TPS)**
- **7 Lean Principles**
- **Eliminate Waste (Muda, Mura, Muri)**
- **Amplify Learning**
- **Decide as Late as Possible**
- **Deliver Fast**
- **Empower the Team**
- **Build Integrity In**
- **See the Whole**
- **Lean vs Agile**