

Lean에 대해 알아보자

@pineoc

이윤석

Index

1. Lean이란?

2. Lean Waste

3. Lean Software Development

4. Lean & Agile

Appendix. Kanban

Lean이란?

Lean이란?

- ~~사전적 정의: 기울다, 기울이다~~
- **Lean Manufacturing**
 - Wiki: [link](#)
 - Lean이라고 불리는 systematic method.
 - **Waste** Minimization
 - Lean Thinking(business methodology)

Lean = Waste Minimization

Lean에서의 Waste

아래의 세 가지 Waste는 TPS의 key concept이다.

TPS = Toyota Production System

- Muda(Uselessness): 낭비, 비 부가가치 활동
- Mura(Unevenness): 공평, 흐름의 가변성
- Muri(Overburden): 과중한 부담

Lean from TPS -

Toyota Production System

- An integrated socio-technical system, developed by Toyota, that comprises its management philosophy and practices.
- [Wiki Link](#)

Lean에서의 Waste

Muda: 낭비, 비 부가가치 활동을 의미함

- * Transport
- * Inventory
- * Motion
- * Waiting
- * Over-production
- * Over-processing
- * Defects

Lean에서 Waste

Mura: Unevenness

- lack of uniformity, irregularity, inequality ...
- 결과물의 품질이 일정(균일)하지 못한 것
- JIT(Just In Time) 방식으로 해결
 - Keeping little or no inventory
 - FIFO(First In First Out)

Lean에서 Waste

Muri: Overburden

- perforce, beyond one's power, immoderation ...
- standardized work(표준화된 작업)으로 개선

So, Waste는 어떻게 제거하는가

Waste는 어떻게 제거하는가

Waste(Muda)의 각 해결방법(Countermeasures)

- * Transport - Value Stream Mapping, Continuous Flow
- * Inventory - JIT, Continuous Flow, Kanban(Pull System)
- * Motion - Value Stream Mapping
- * Waiting - Continuous Flow
- * Over-production - Takt Time, Kanban(Pull System)
- * Over-processing - Kaizen
- * Defects - Standardized Work

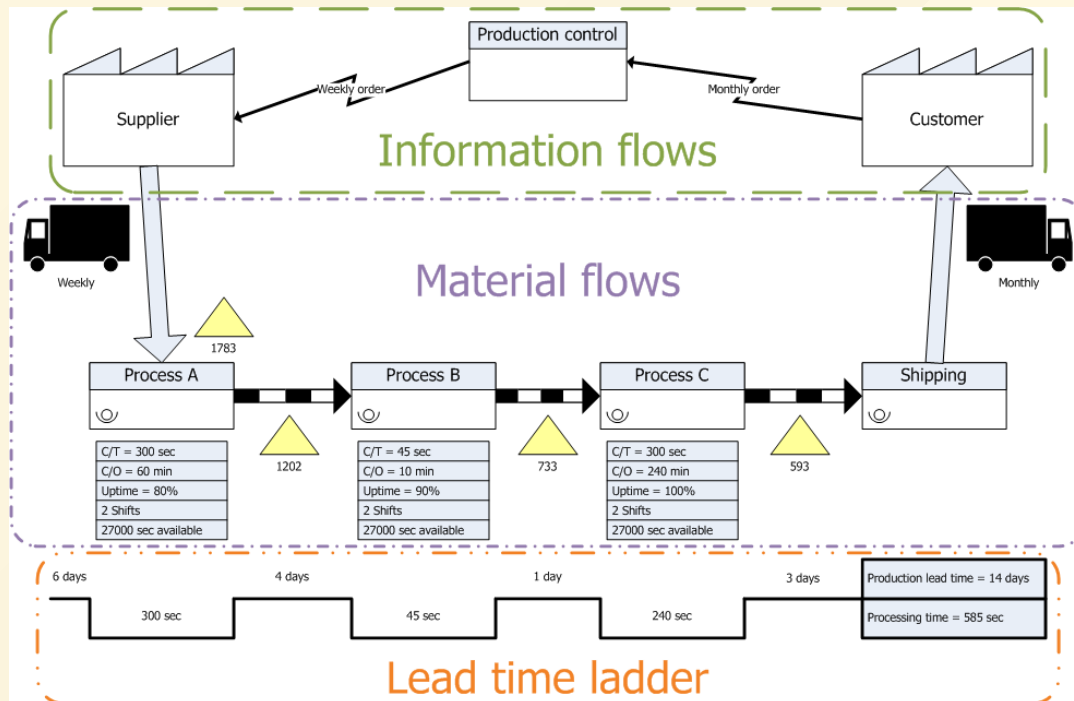
각 해결방법 설명

(각각 따로 정리해야함)

- * Value Stream Mapping
- * Continuous Flow
- * JIT(Just In Time)
- * Kanban(Pull System)
- * Takt Time
- * Kaizen
- * Standardized Work

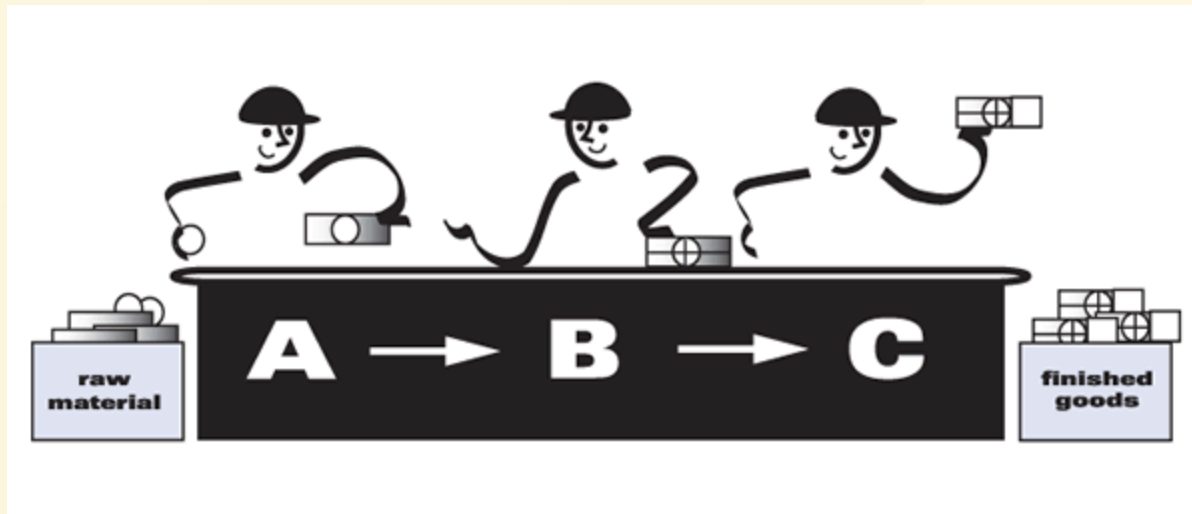
Value Stream Mapping

Analyzing the current state and designing a future state for the series of events that take a product or service from its beginning through to the customer.



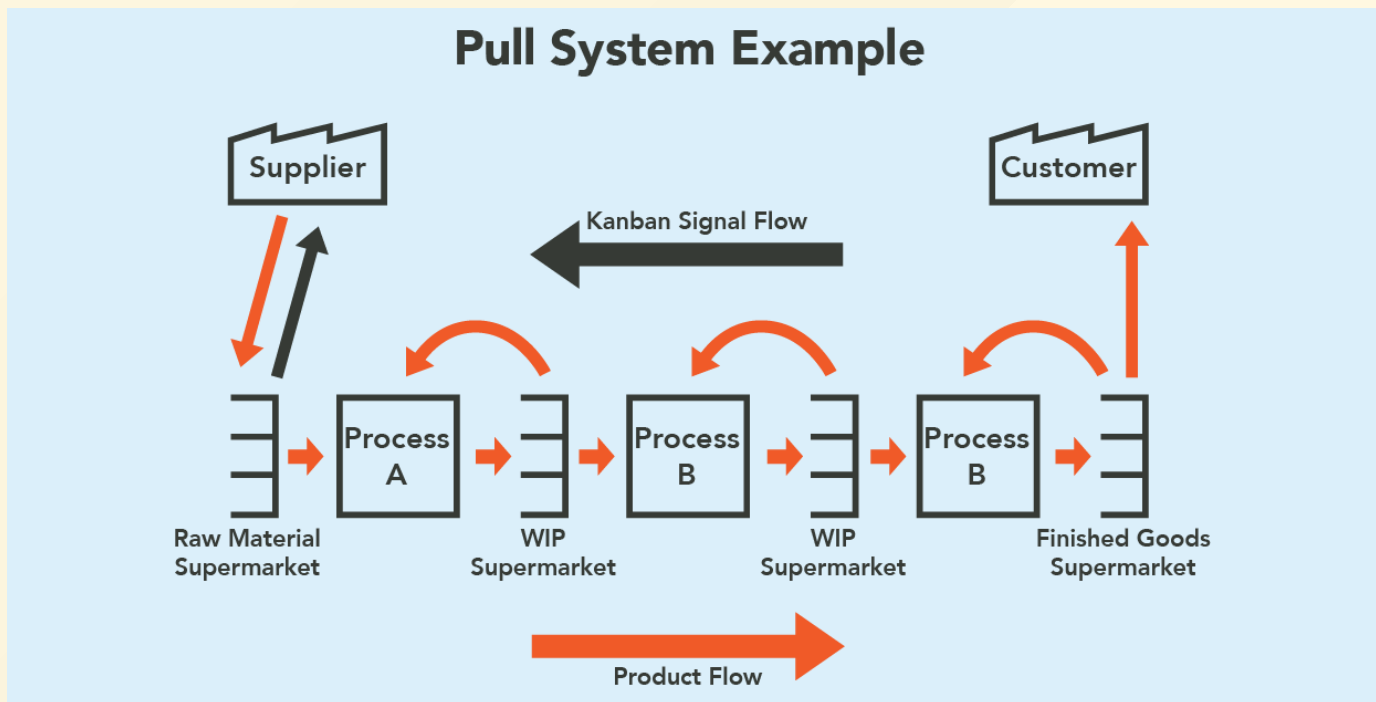
Continuous Flow

Work in progress smoothly flows through production with minimal buffers between steps of the manufacturing process



JIT(Just In Time)

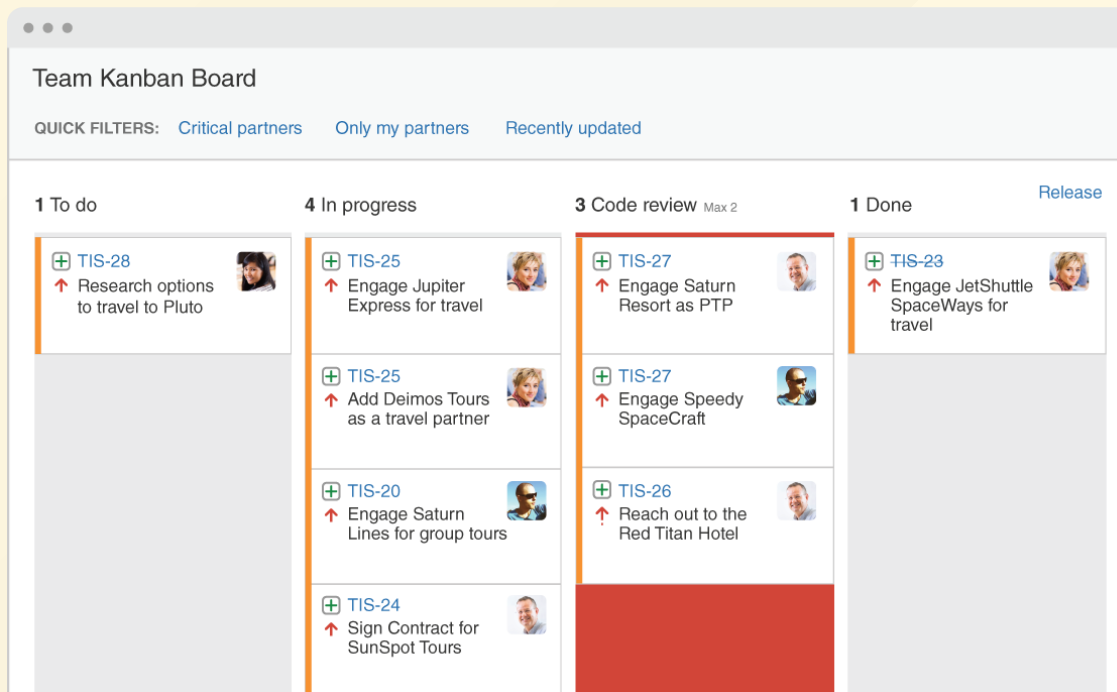
Reducing flow times within production system as well as response times from suppliers and to customers.



Kanban(Pull System)

lean method to manage and improve work across human systems.

(Appendix로 추가 설명할 예정)



Takt Time

The available production time divided by customer demand.

= 사용 가능한 생산 시간을 고객 수요로 나눈 값.

Kaizen (개선)

"improvement", Continuous Improvement

Standardized work

Create work instructions that provide a consistent method of manufacturing the part.

Lean Software Development

Lean Software Development

“ Lean software development(LSD) is a translation of **lean manufacturing** principles and practices to the **software development domain**. ”

목표:

시스템의 **낭비**를 줄이고, 고객에게 더 높은 가치를 만든다.

Lean SD, Waste

Lean Philosophy에서의 8 Waste

- 1 Partially done work
- 2 Extra processes
- 3 Extra features
- 4 Task switching
- 5 Waiting
- 6 Motion
- 7 Defects
- 8 Management activities

Lean SD, Waste

소프트웨어에서의 9 Waste

- 1 Building the wrong feature or product
- 2 Mismanaging the backlog
- 3 Rework
- 4 Unnecessarily complex solutions
- 5 Extraneous cognitive load
- 6 Psychological distress
- 7 Waiting/multitasking
- 8 Knowledge loss
- 9 Ineffective communication.

Lean SD, 7 Principles

- Eliminate Waste
- Amplify learning
- Decide as late as possible
- Deliver as fast as possible
- Empower the team
- Build integrity in
- See the whole

Lean SD, Principles - Eliminate Waste

- Value stream mapping을 통해 waste를 찾는다
- (앞에 있던 9개의 waste 리스트)
- waste가 있는 지점을 찾고 제거한다
- 핵심 공정만 남을 때까지 iteratively하게

Lean SD, Principles - Amplify learning

Software development is a continuous learning process based on iterations when writing code.

- 잦은 반복(iteration)과 고객의 피드백을 통해 경험과 지식을 창출하라
- 잦은 반복
 - 요구사항 -> 설계 -> 개발 -> 테스트
- 방식
 - Pair Programming, Code Review ...

Lean SD, Principles -

Decide as late as possible

- 예측이 아닌 **사실을 기반**으로 결정을 할 수 있을 때까지 확정을 늦춰라
- 왜?
 - 결정을 미루라는 말이 아님, 정보를 최대한 많이 가지고 있을 때의 결정이 유리하기 때문
 - 개발 환경 = 예측이 불가능한 상황
 - 결정은 하되 변화를 수용할 수 있어야 함

Lean SD, Principles -

Deliver as fast as possible

- 고객의 확실한 요구사항을 파악하기 위해 제품을 가능한 빨리 인도하라
- 품질의 내재화가 필수 조건이다
- 필요조건
 - 낭비의 제거, 품질의 내재화
 - 일의 양 제한(WIP 제한)

Lean SD, Principles - Empower the team

- Respect People
- 사람을 대체 가능한 인력으로 다루지 말라
- 사람을 신뢰하고, 전문 기술을 가질 수 있도록 인재를 육성하라

Lean SD, Principles - Build integrity in

- 결함을 예방하는 테스트를 통해 코드의 품질을 내재화하라
- 실행방법: TDD, Continuous Integration, Pair Programming...
- 빅뱅 통합보다 지속적 통합을 하라
- 자동화된 단위 테스트와 인수 테스트 작성을 하라

Lean SD, Principles - See the whole

- 부분 최적화가 전체를 최적화하지 않을 수 있다
- 문제의 근본 원인을 찾아 전체를 최적화하라

“ Think big, act small, fail fast; learn rapidly ”

Lean Software Practices -

Waste를 줄이기 위한 방법들

- * Seeing waste
- * Value stream mapping
- * Set-based development
- * Pull systems(Kanban)
- * Queuing theory
- * Motivation
- * Measurements
- * TDD(Test Driven Development)

Lean & Agile(Scrum)

Lean & Agile(Scrum)

Kanban