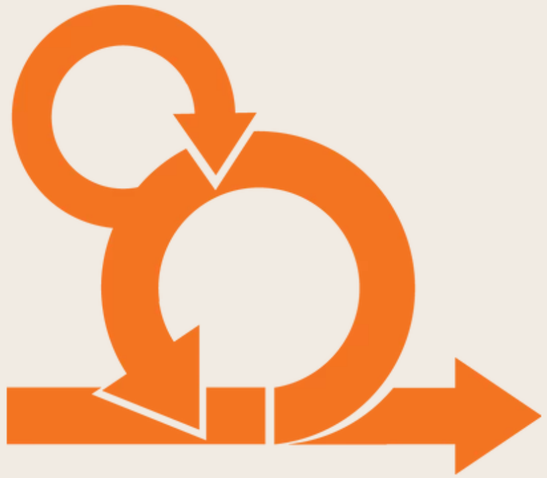
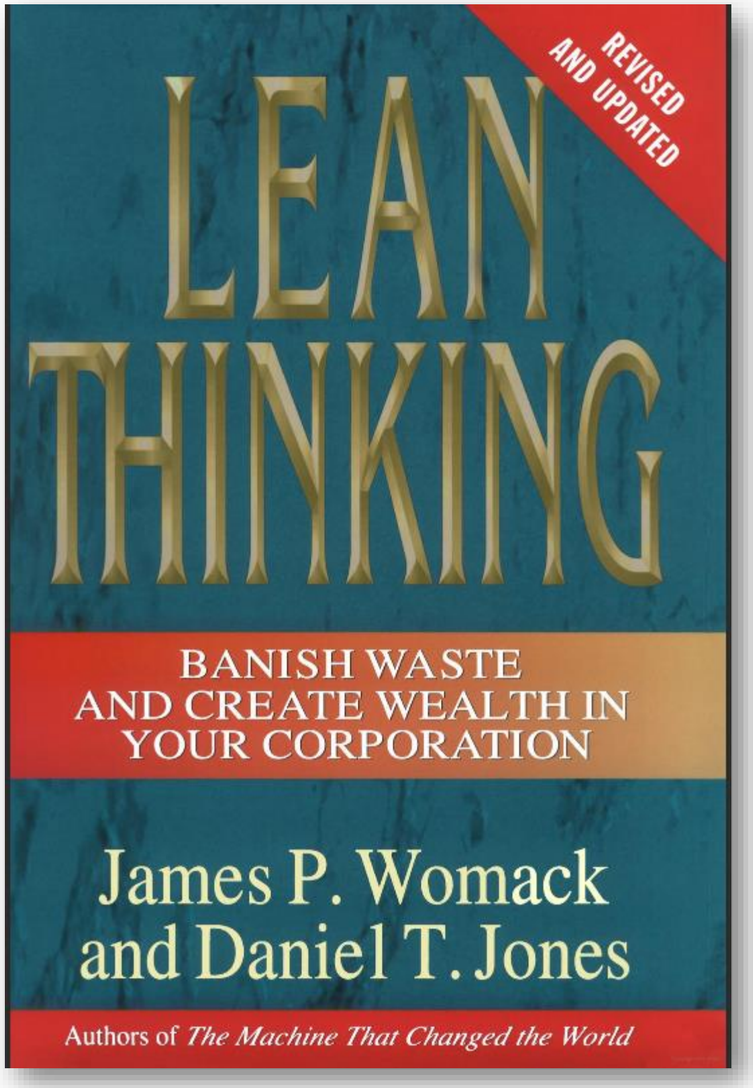
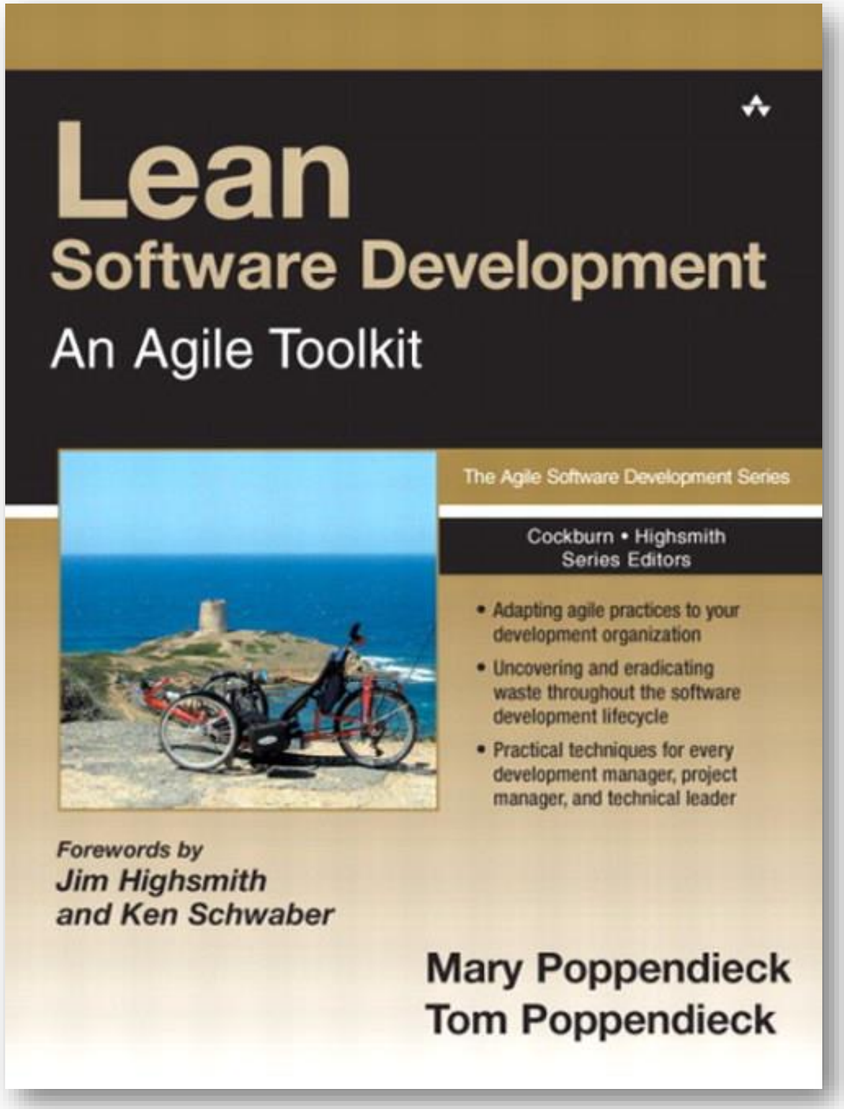
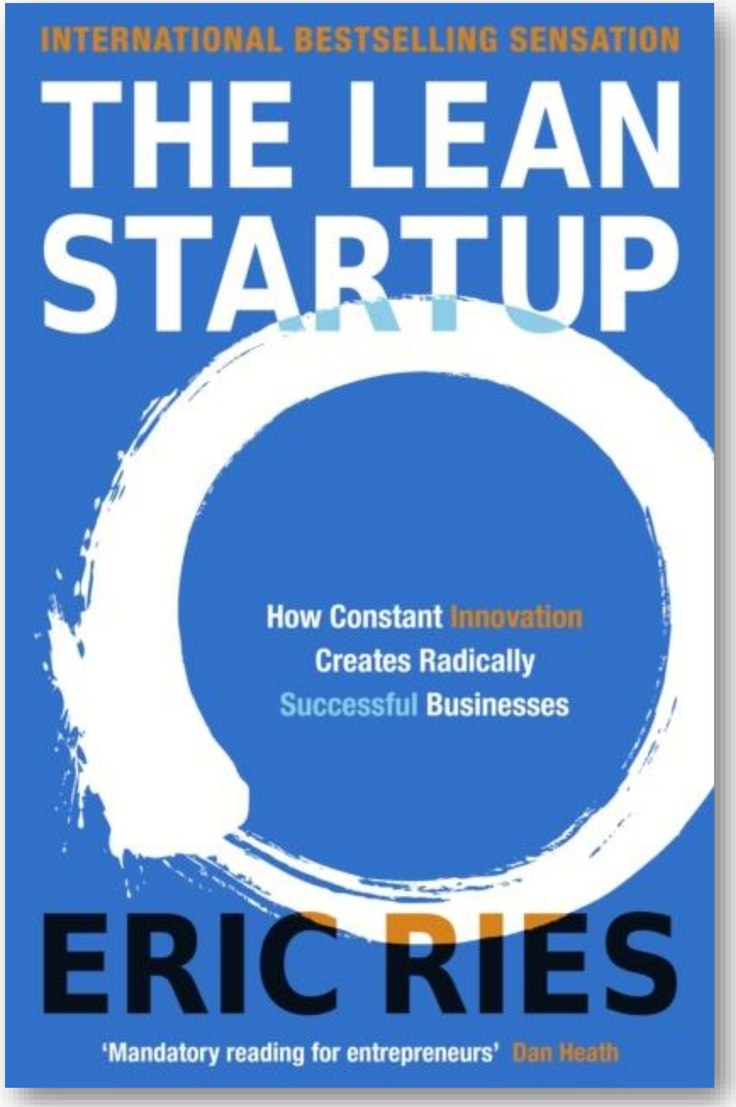


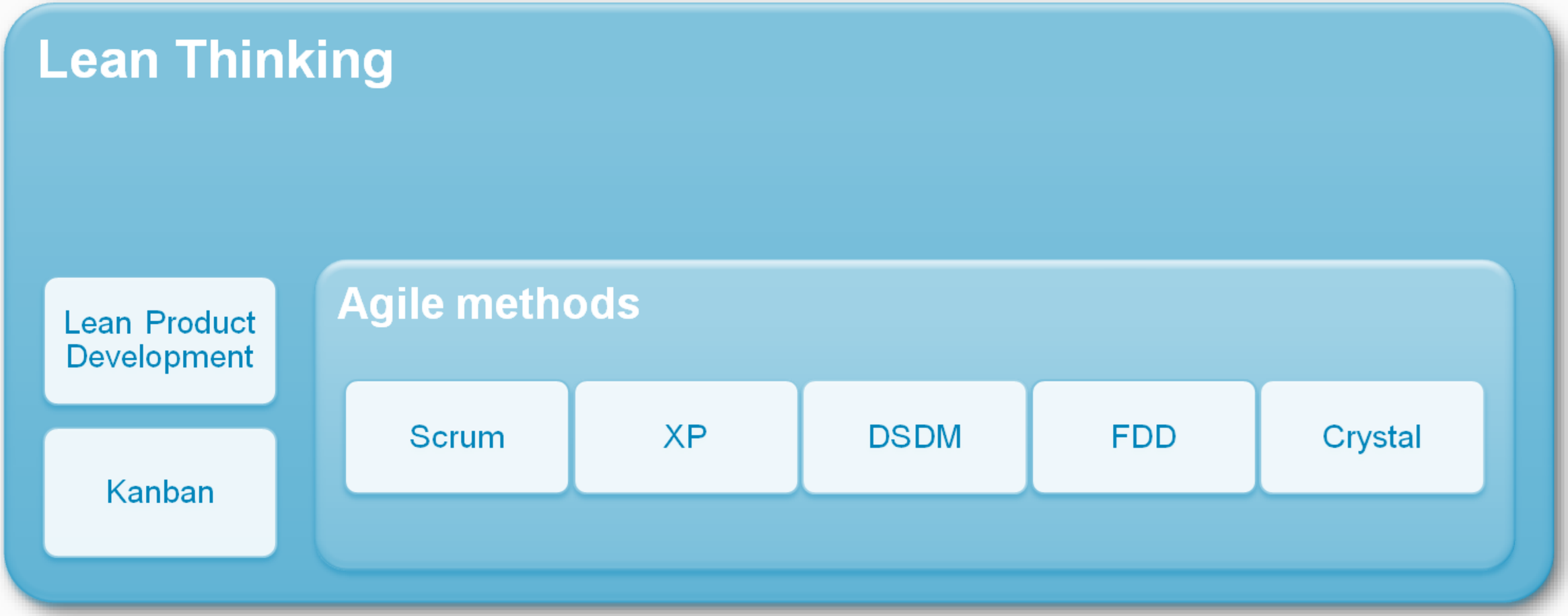
AGILE SOFTWARE DEVELOPMENT

Lean Software Development


REVISED AND UPDATED
LEAN THINKING
BANISH WASTE AND CREATE WEALTH IN YOUR CORPORATION
James P. Womack and Daniel T. Jones
Authors of The Machine That Changed the World


Lean
Software Development
An Agile Toolkit
The Agile Software Development Series
Cockburn • Highsmith
Series Editors
• Adapting agile practices to your development organization
• Uncovering and eradicating waste throughout the software development lifecycle
• Practical techniques for every development manager, project manager, and technical leader
Forewords by
Jim Highsmith
and Ken Schwaber
Mary Poppendieck
Tom Poppendieck


INTERNATIONAL BESTSELLING SENSATION
THE LEAN STARTUP
How Constant Innovation Creates Radically Successful Businesses
ERIC RIES
'Mandatory reading for entrepreneurs' Dan Heath


Lean Thinking
Lean Product Development
Kanban
Agile methods
Scrum
XP
DSDM
FDD
Crystal

Agile

- Adaptive to change
- Shorter planning and commitment cycles
- Focus on collaboration and interaction

Lean

- System view of value stream
- Identify ways to eliminate waste
- Limit work queues

Agile & Lean Commonalities

- Improve quality
- Amplify learning
- Continuously improve
- Empower people

"Think big, act small, fail fast; learn rapidly"

A Lean History

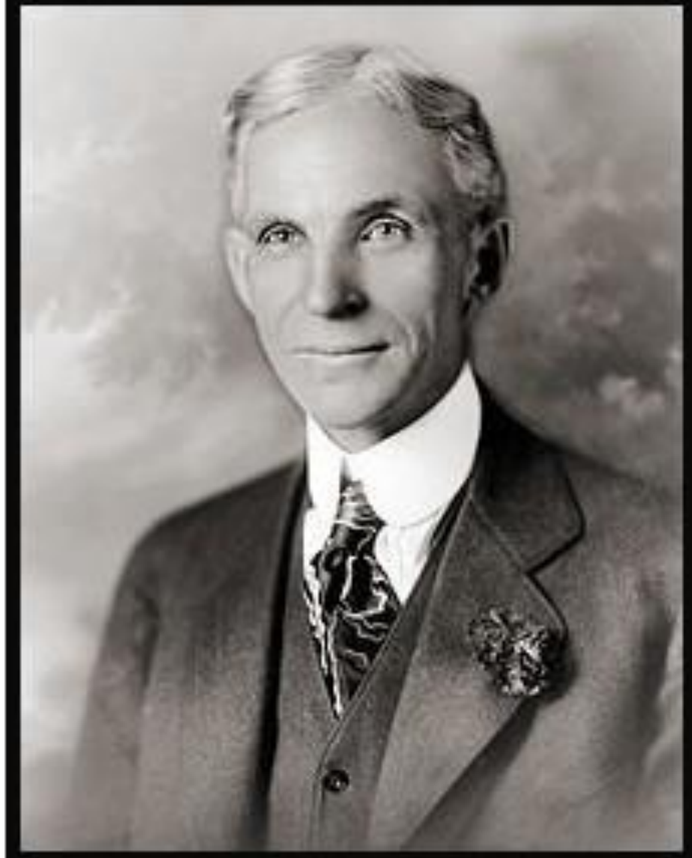
- Lean is a *manufacturing* & *production* practice that considers the expenditure of resources for any goal other than the creation of **value for the end customer** to be wasteful, and thus a target for elimination

- "**value**" is defined as any action or process that a customer would be willing to pay for

A Lean History

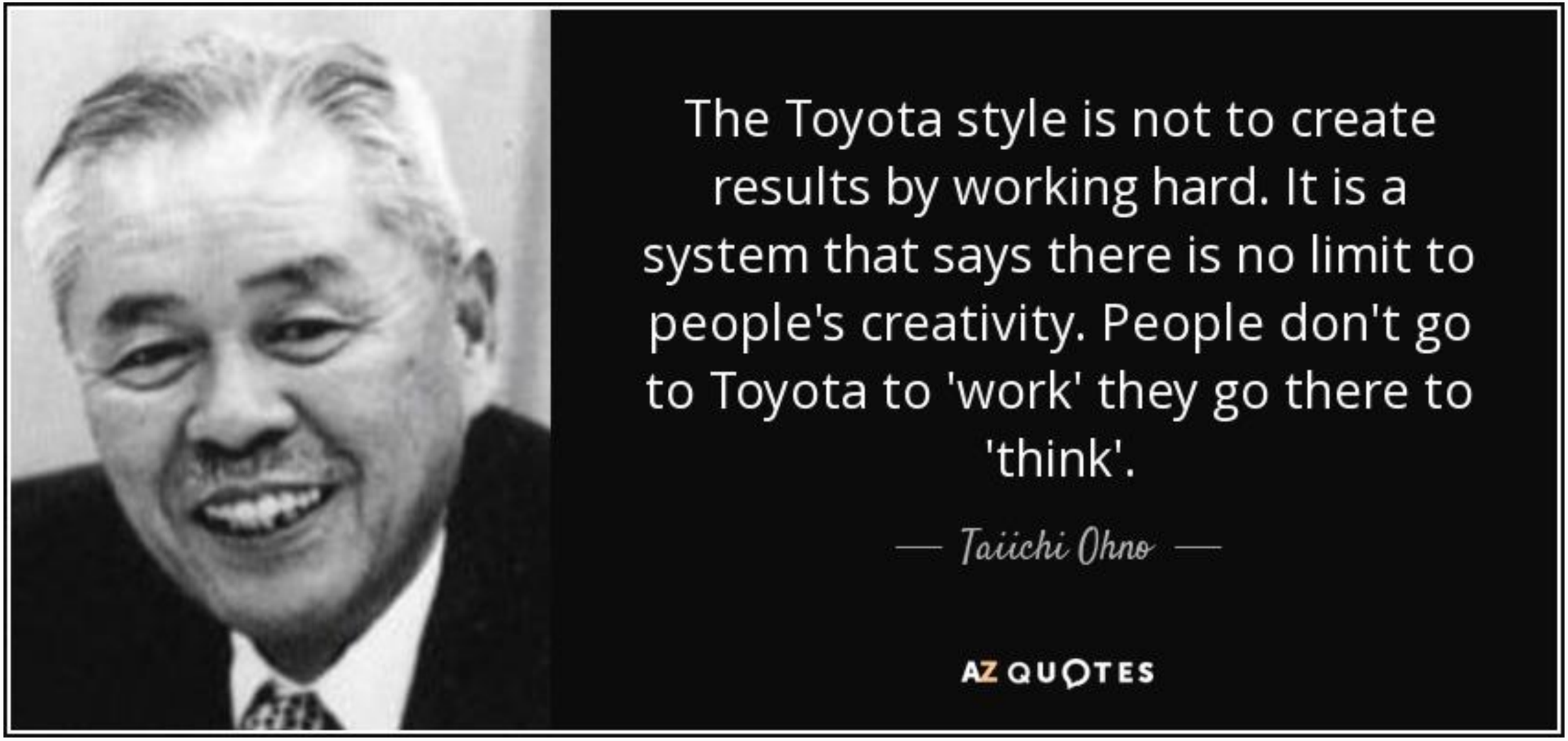
- Lean is centered around preserving value with less work
- Lean manufacturing is based on
  - optimizing flow,
  - increasing efficiency,
  - decreasing waste,
  - using empirical methods to decide what matters, rather than uncritically accepting pre-existing ideas
- Toyota was a leader in implementing lean practices in the 80s


Taiichi Ohno
Toyota Production System



Any customer can have a car painted any colour that he wants so long as it is black.

(Henry Ford)


The Toyota style is not to create results by working hard. It is a system that says there is no limit to people's creativity. People don't go to Toyota to 'work' they go there to 'think'.
— Taiichi Ohno —
AZ QUOTES

Toyota Production System :

How could Toyota make cars in small quantities but keep them as inexpensive as mass-produced cars?

“Just-in-time” manufacturing

“Don't decide what to manufacture until you have a customer order; then make it as fast as possible”

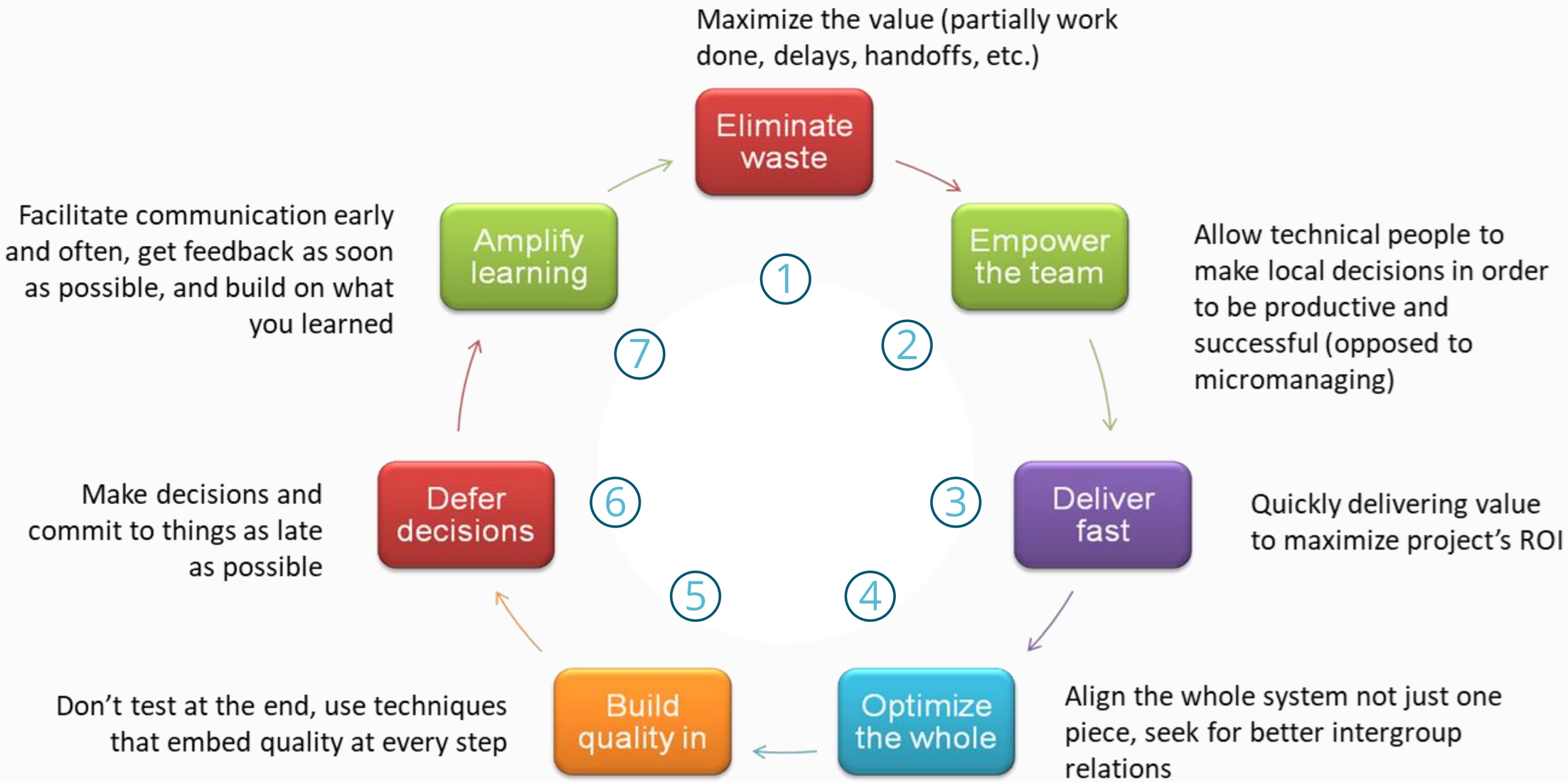
Case Study: Statewide Automated Child Welfare Information System (SACWIS)

- **Florida:** started in 1990, estimated 8 years and $32 million
  - In 2002 Florida spent $170 million and estimated to be completed in 2005 with $230 million
- **Minnesota:** started in 1999
  - completed in 2000 at cost of $1.1 million

- Why? Standardized infrastructure, minimized requirements, team of 8 capable people

Source: Standish Group

Lean Principles for Software Development



Lean Principles are… just Principles

- **Eliminate waste** does not mean throw away all documentation.
- **Empower the team** does not mean abandon leadership.
- **Deliver as fast as possible** does not mean rush and do sloppy work.
- **See the whole** does not mean ignore the details.
- **Build integrity in** does not mean big, upfront design.
- **Decide as late as possible** does not mean procrastinate.
- **Amplify learning** does not mean keep on changing your mind.

1. Eliminate waste

If a development cycle has collected requirements in a book gathering dust, that is waste

If developers code more features than are immediately needed, that is waste

Whatever gets in the way of rapidly satisfying a customer need is waste.

Handing off development from one group to another is waste

The Seven Wastes of ~~Manufacturing~~ Software Development

- ~~Inventory~~ Partially Done Work
- ~~Extra Processing~~ Extra Processes
- ~~Overproduction~~ Extra Features
- ~~Transportation~~ Task Switching
- Waiting
- Motion
- Defects

Shigeo Shingo, Toyota

Eliminate waste

WASTE

=

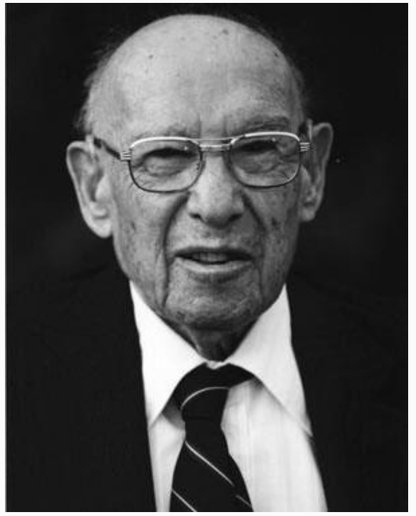
everything your organization does to develop software that is not analysis or

coding.

It is usually easier to see waste in a crisis

Eliminate waste

"There is nothing so useless as doing efficiently that which should not be done at all"



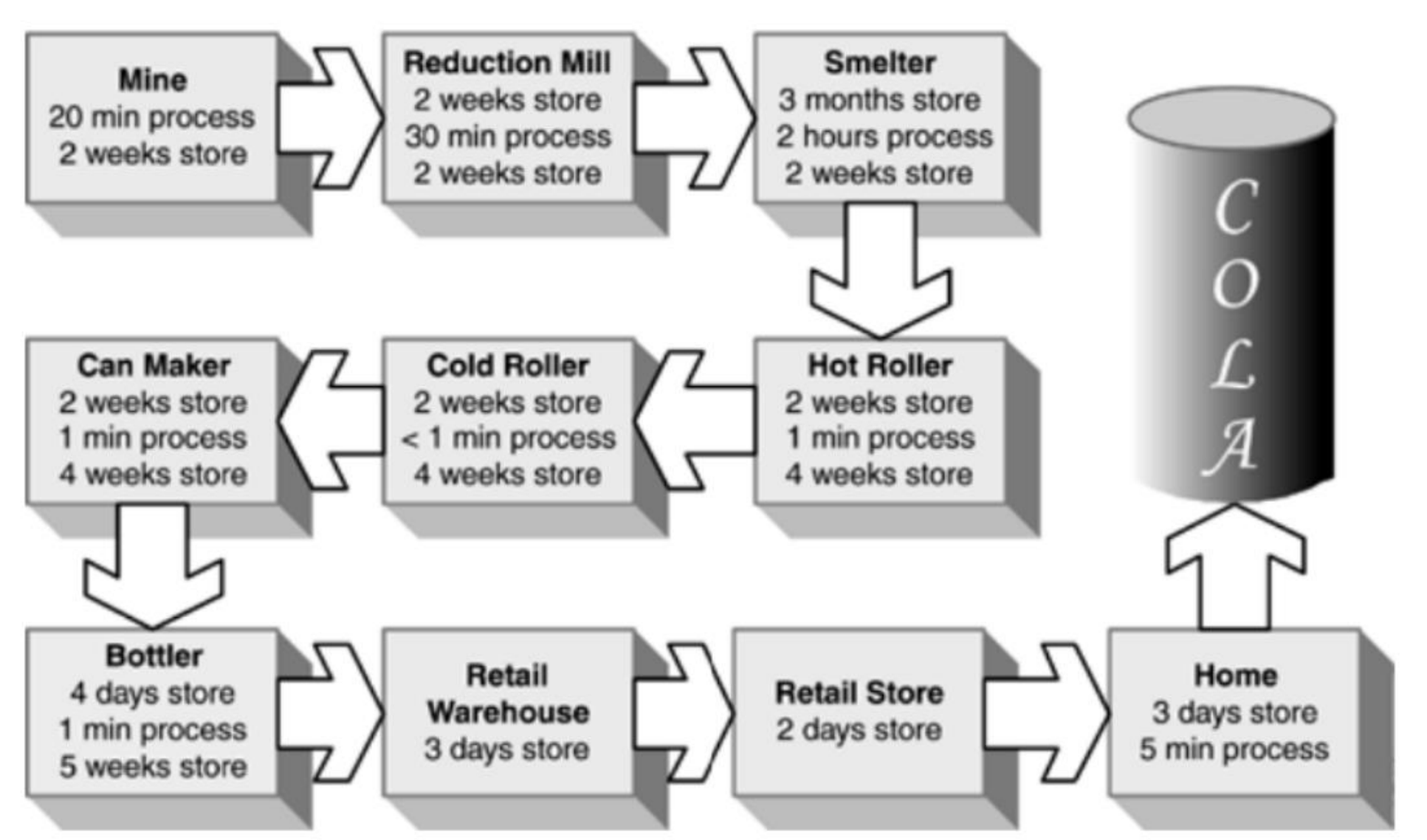
Peter Drucker

Eliminate waste

1. Implementing lean development is learning to see waste.
2. Uncover the biggest sources of waste and eliminate them.
3. Uncover the biggest remaining sources of waste and eliminate them.
4. Do it again.

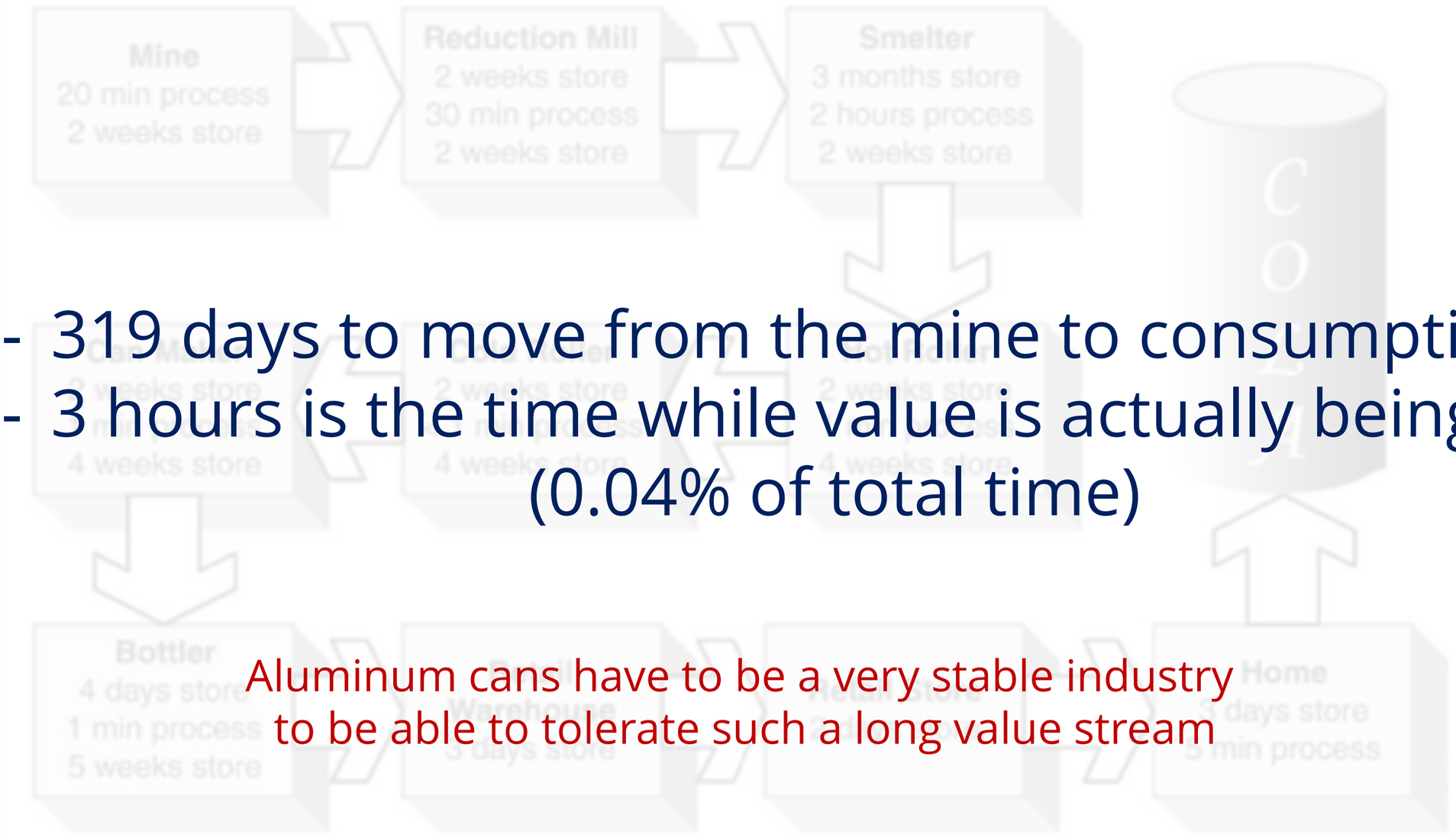
After a while, even things that seem essential can be gradually eliminated

Value Stream for Cola Cans



James P. Womack, Daniel T. Jones (1997)

Value Stream for Cola Cans



- 319 days to move from the mine to consumption
- 3 hours is the time while value is actually being added (0.04% of total time)

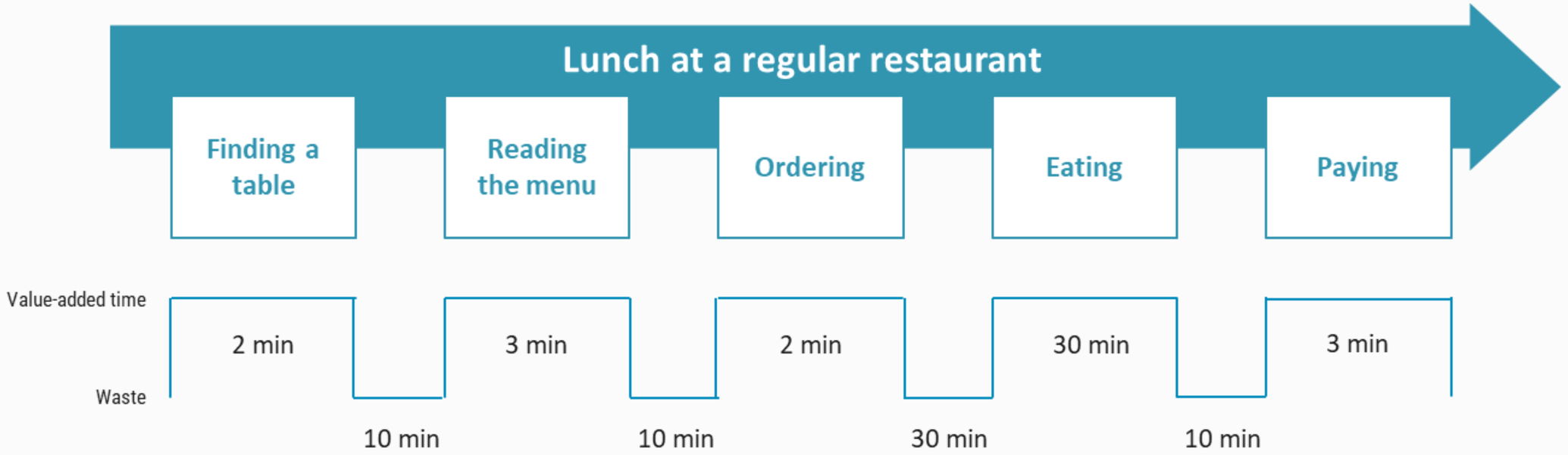
Aluminum cans have to be a very stable industry
to be able to tolerate such a long value stream

James P. Womack, Daniel T. Jones (1997)

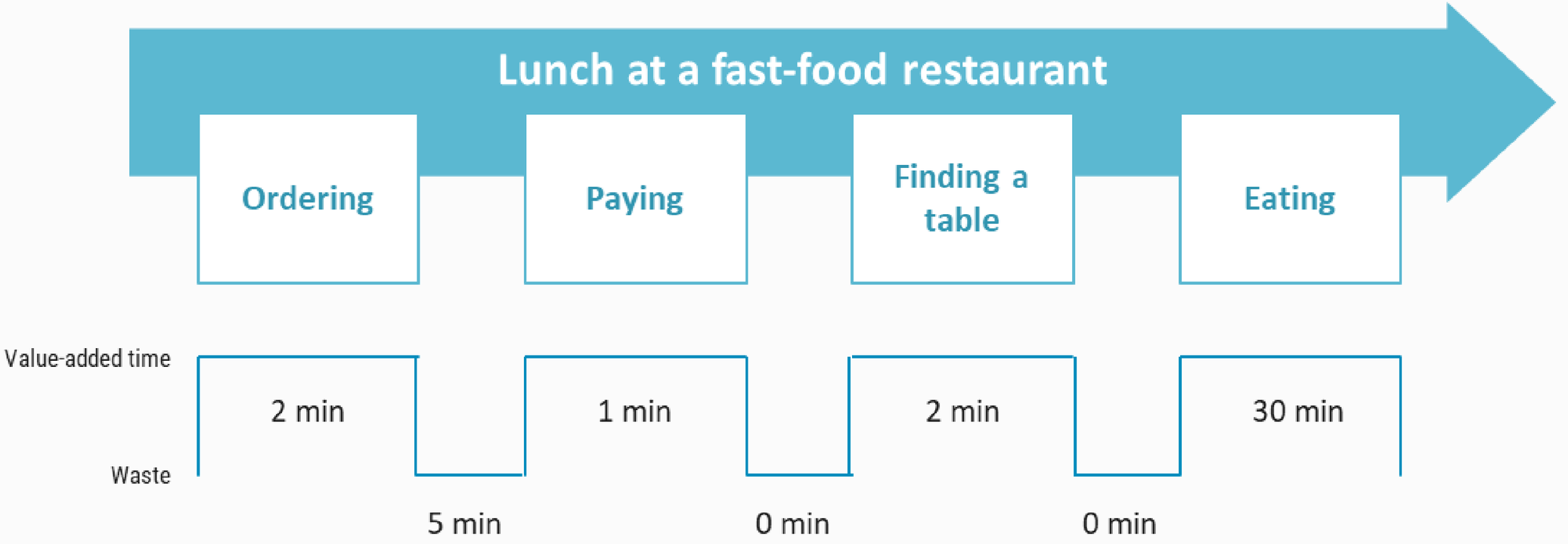
…not working for personal computers

Michael Dell

“8 days of inventory – competitors 40 days.
If Intel comes out with a new chip, I am going to get that to the market 32 days sooner."

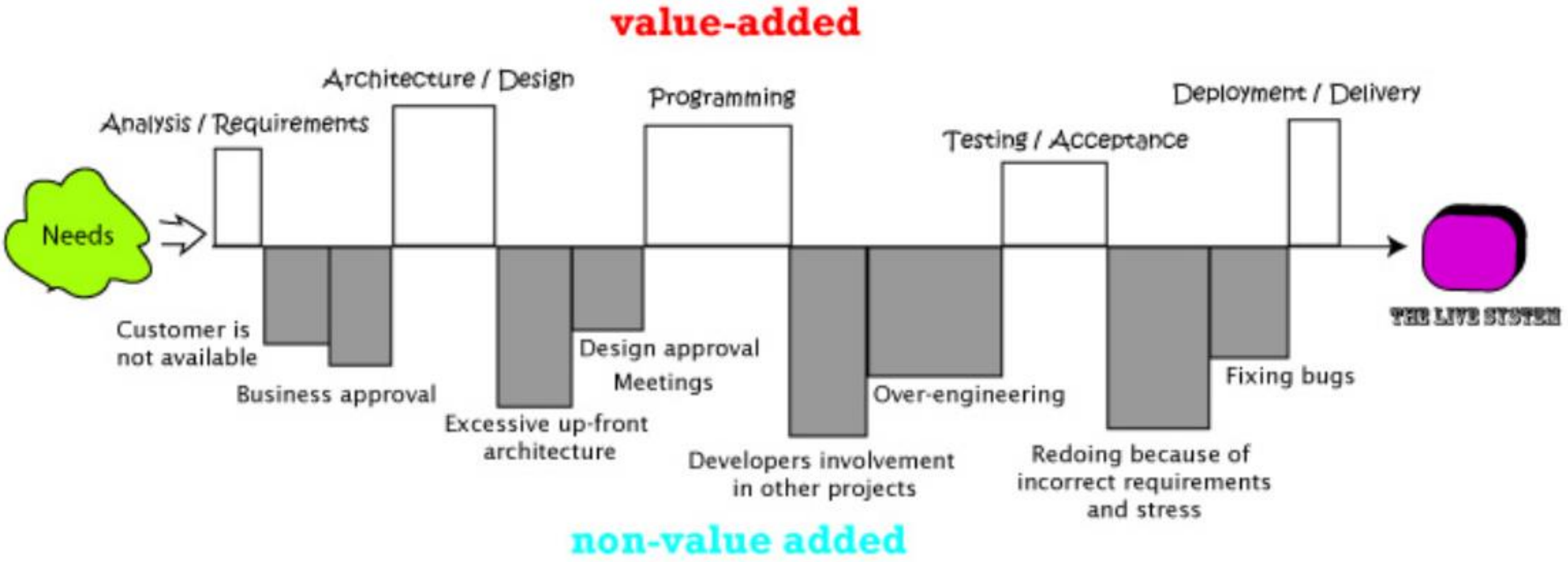


Total cycle time = 100 min
Value-added time = 40 min
Cycle efficiency = 40 / 100 = 40%



Total cycle time = 40 min
Value-added time = 35 min
Cycle efficiency = 35 / 40 = 88%

Eliminate waste



Eliminate waste

How to eliminate waste:

- Make a list of the 10 or 15 most important activities in your organization
- Rate 1-5 (1 customer do not care about , 5 customers value it highly)
- Develop a plan to cut those with 1 or 2 points

Eliminate waste

How to eliminate waste:

Develop a value stream map

Take the biggest cause of delay and plan to cut it in half

Eliminate waste

How to eliminate waste:

Seven meetings talk about the wastes in software development:

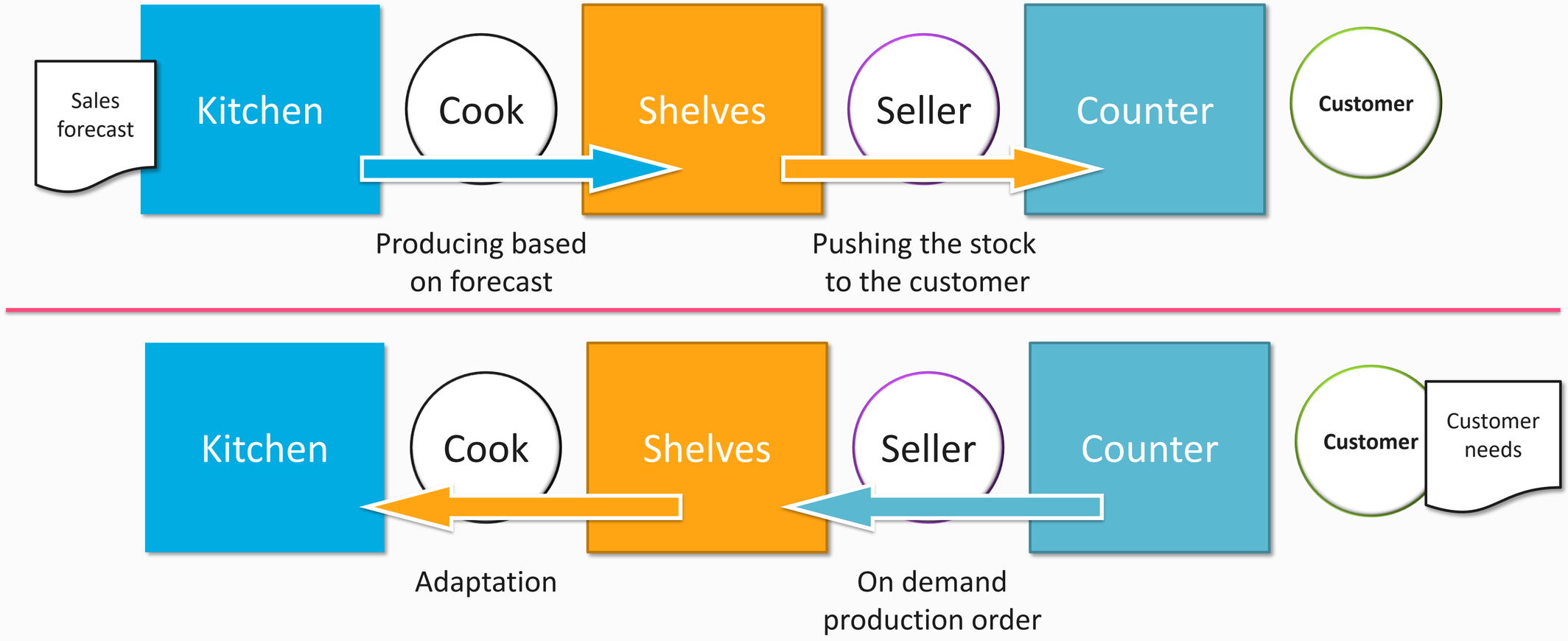
- Do you agree that this is waste? Why?
- How much time it consumes in avg / week
- What can we do to reduce that time

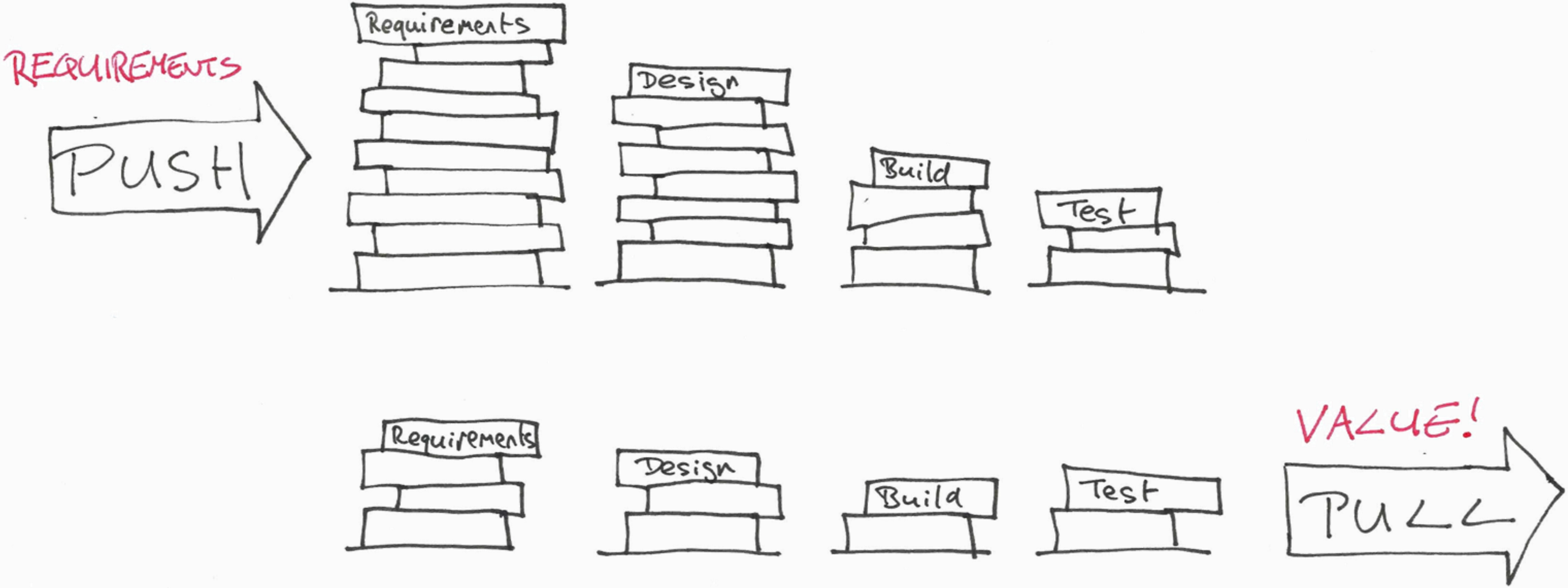
3. Deliver as fast as possible

- Customers like rapid delivery
- Rapid delivery means less time for customers to change their minds
- In-process, or partially done work can have undiscovered defects
- *Deliver as fast as possible* complements *decide as late as possible:* the faster you can deliver, the longer you can delay decisions.

3. Deliver as fast as possible

Push versus Pull




REQUIREMENTS
PUSH!
Requirements
Design
Build
Test
Requirements
Design
Build
Test
VALUE!
PULL

6. Decide as late as possible

In an evolving market, keeping design options open is more valuable than committing early.

How to avoid change penalties?

- Traditional: make the right design decision in the first place and avoid the need to change later

- Lean: Don't make irreversible decisions in the first place; delay design decisions as long as possible, and when they are made, make them with the best available information to make them correctly

6. Decide as late as possible

- The last responsible moment:
  - delay commitment until the last responsible moment, that is, the moment at which failing to make a decision eliminates an important alternative.