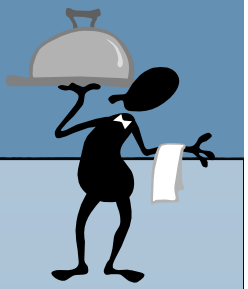
l e a n
software development

Lean Software Development

An Introduction

mary@poppendieck.com Mary Poppendieck www.poppendieck.com

What is Lean?

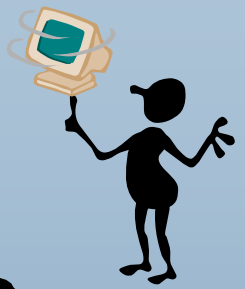


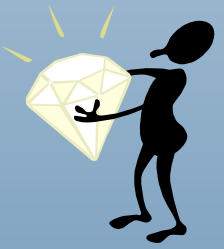
Deliver continually increasing customer value

Expending continually decreasing effort

In the shortest possible timeframe

With the highest possible quality



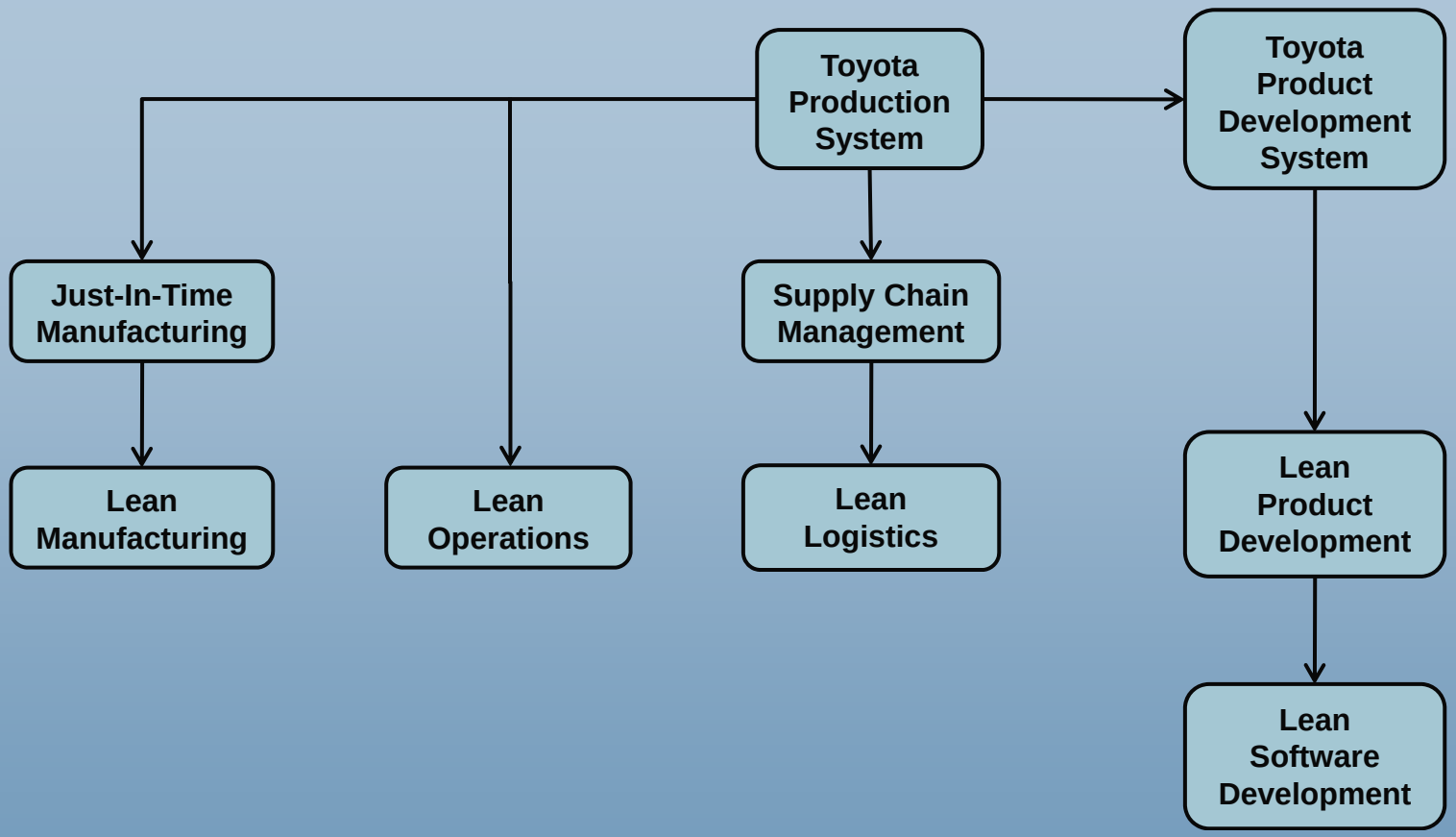


A Journey

Not a Destination

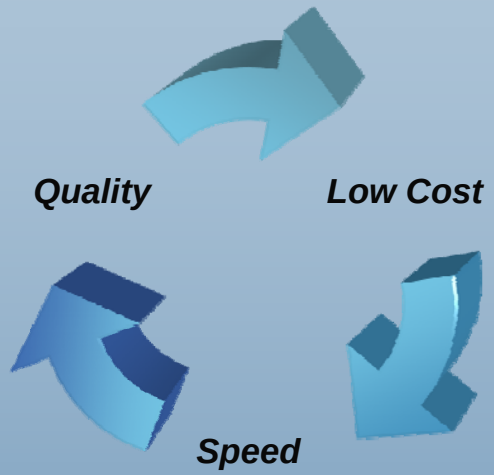
lean

The Toyota Production System Precursor of Lean Thinking



Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



lean

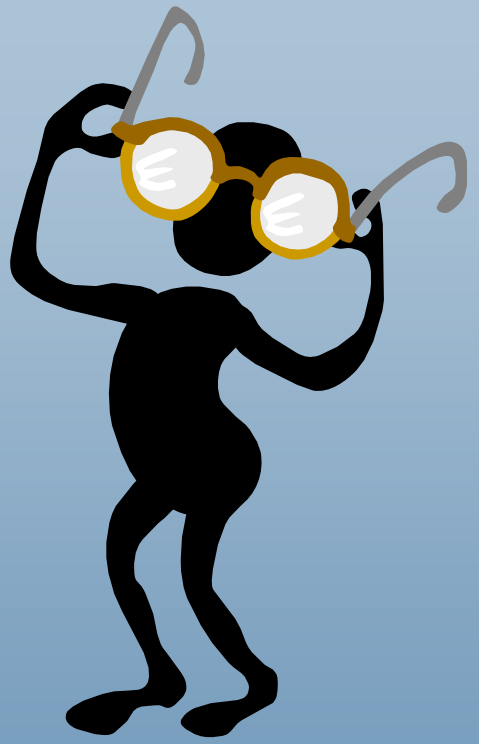
Learn to See Waste

Waste is…

Anything that depletes resources of time, effort, space, or money without adding customer value.

Value is…

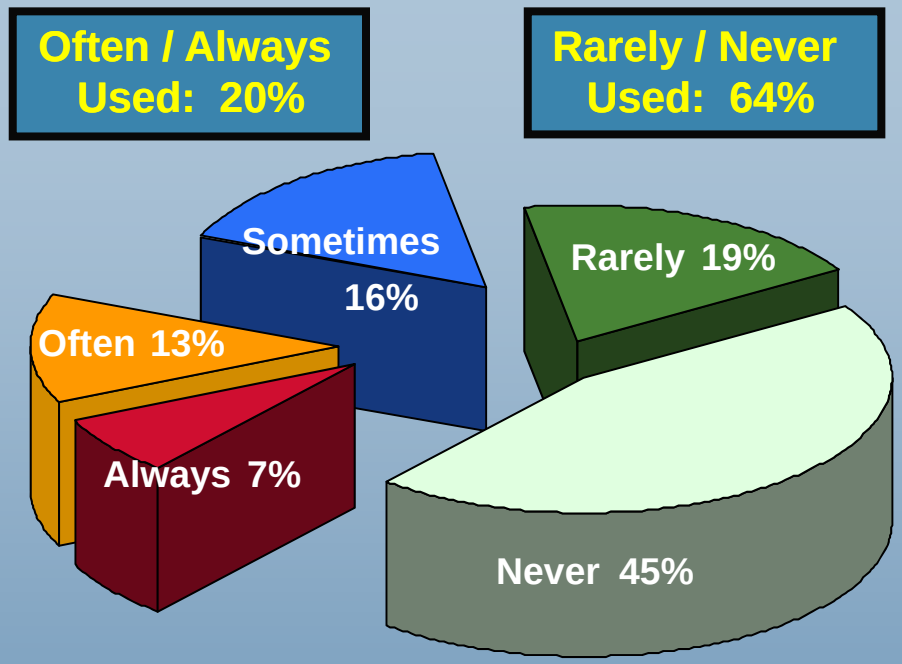
Seen through the eyes of those who pay for, use, support, or derive value from our systems.



Put on Customer Glasses

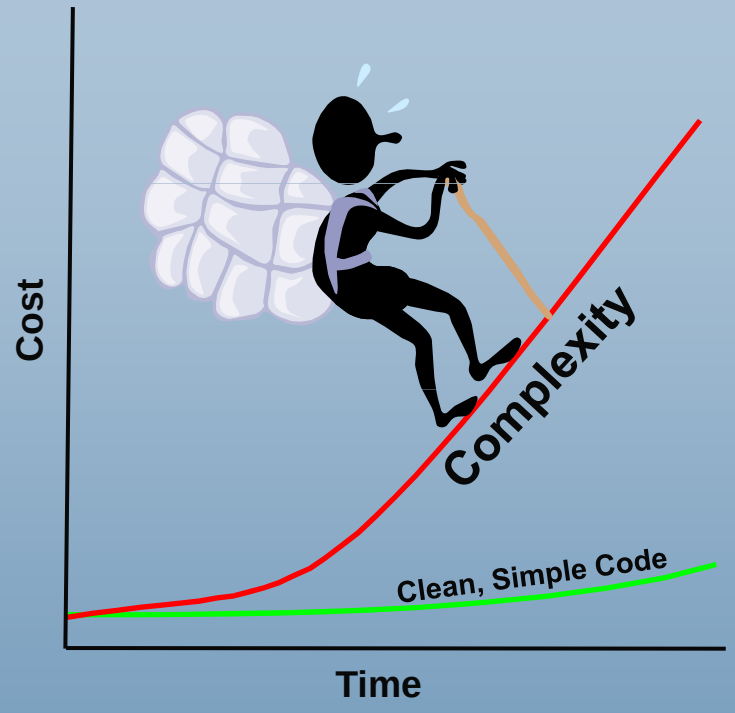
Extra Features are Waste

Features / Functions Used in a Typical System



Standish Group Study Reported at XP2002 by Jim Johnson, Chairman

Cost of Complexity



The Biggest opportunity for increasing Software Development Productivity: Write Less Code!

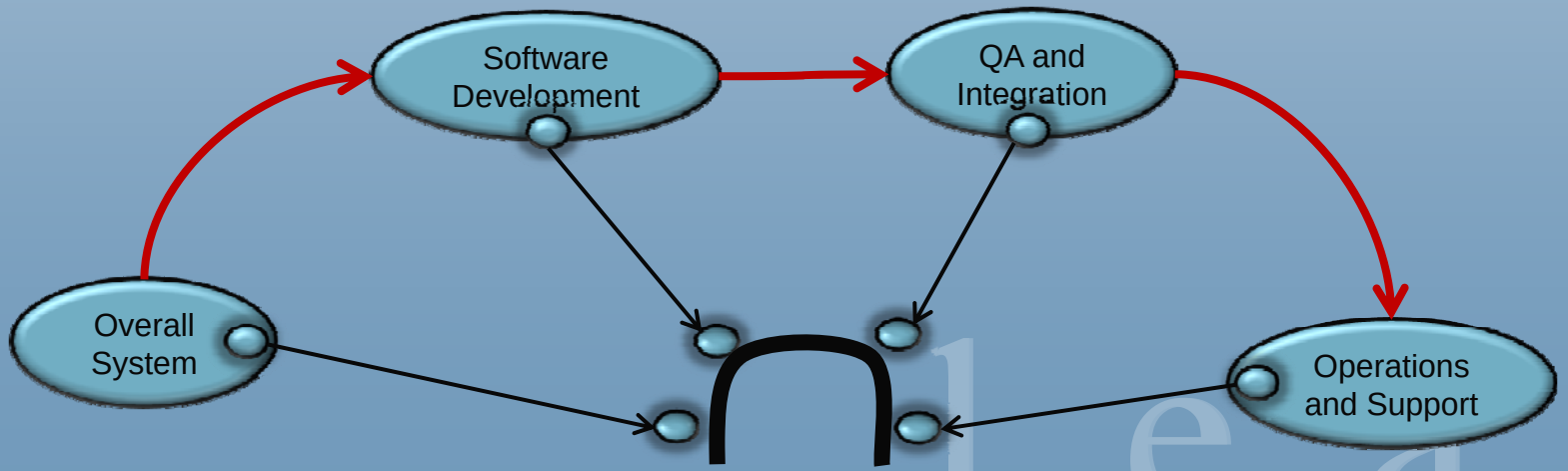
Handovers are Waste

*Alan Ward: Lean Product and Process Development

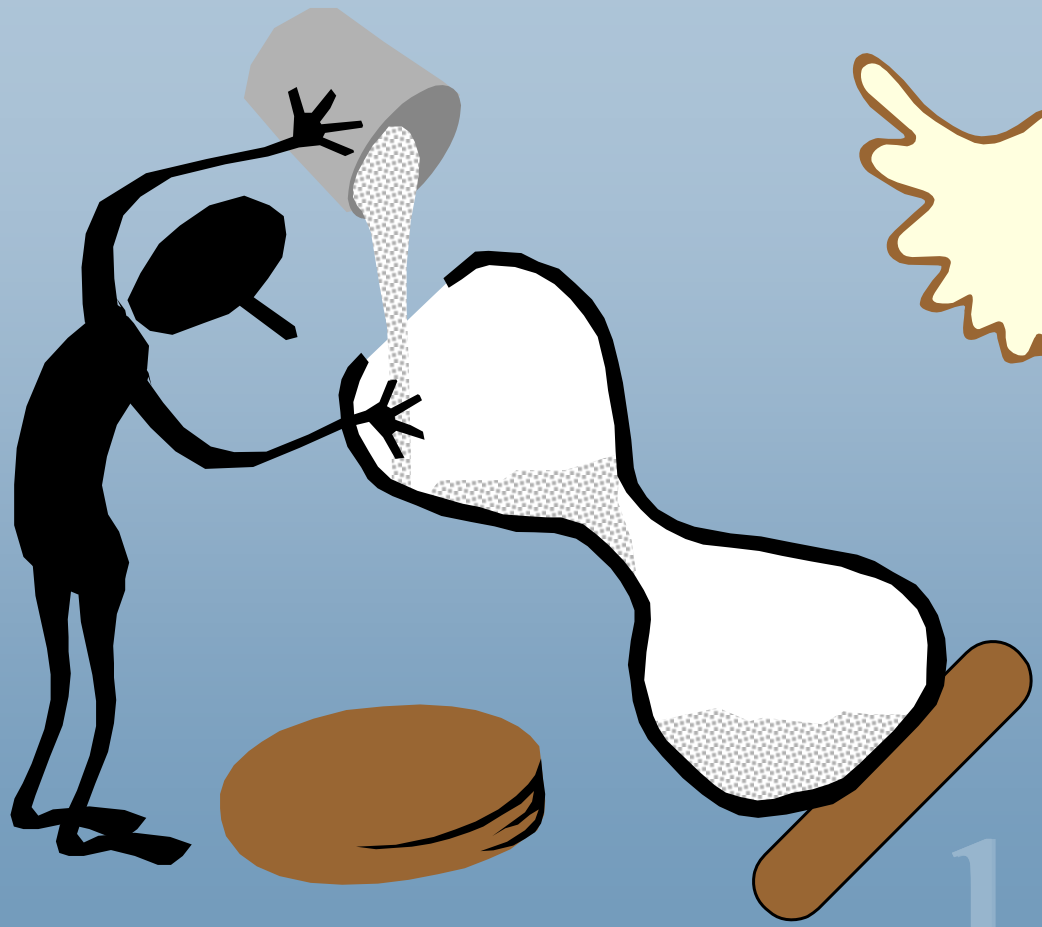
*A handover occurs whenever we separate:**

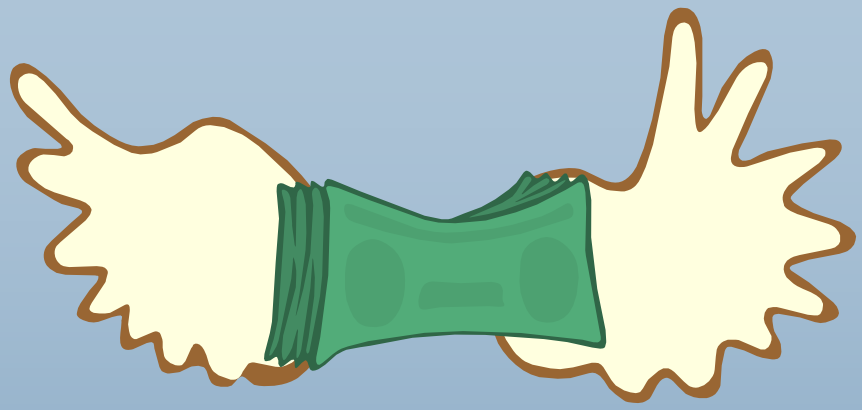
- ✓ Responsibility – What to do
- ✓ Knowledge – How to do it
- ✓ Action – Actually doing it
- ✓ Feedback – Learning from results

Cross Functional Teams

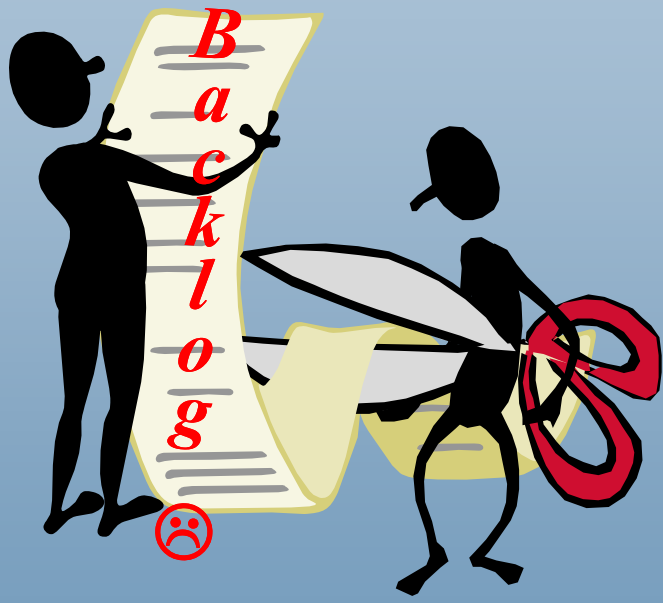


Delays are Waste

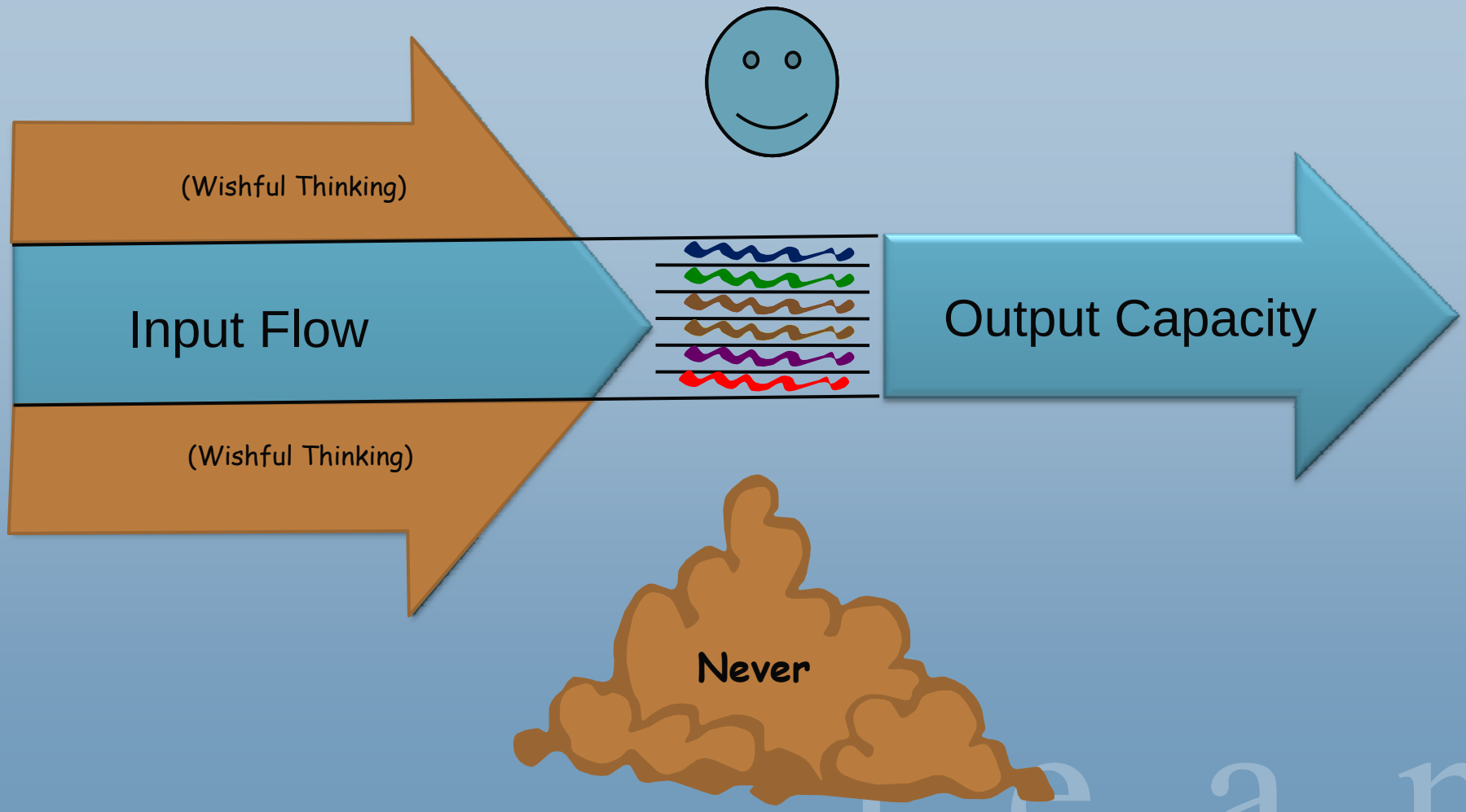




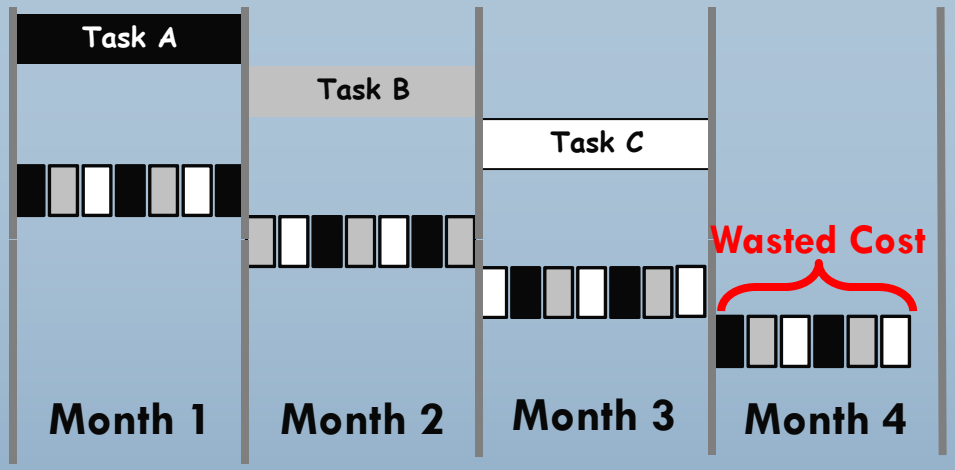
Work in Progress Waste

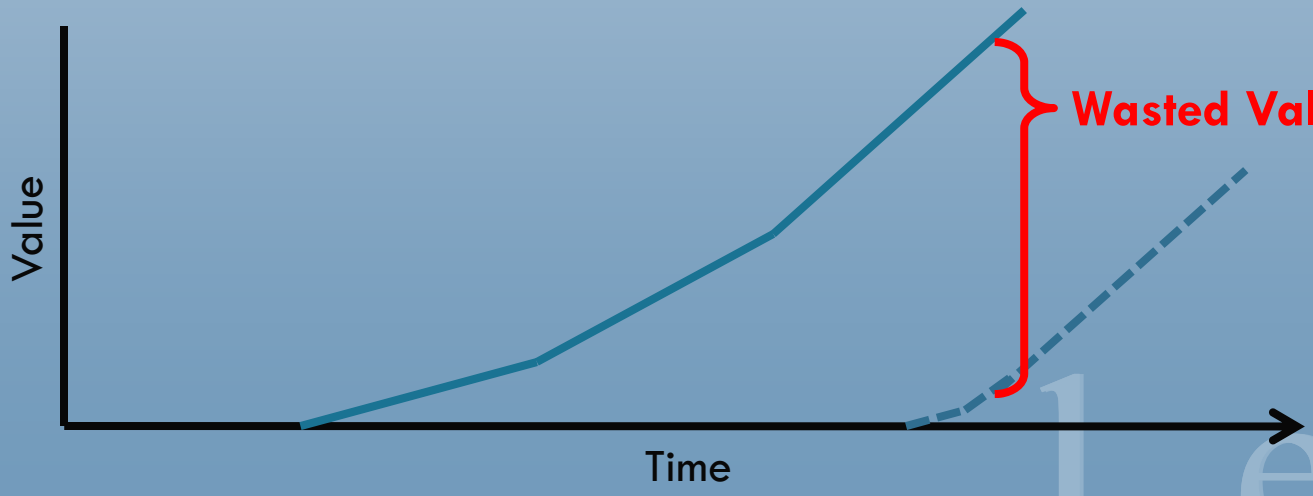


Why do you need a Backlog?



Task Switching is Waste





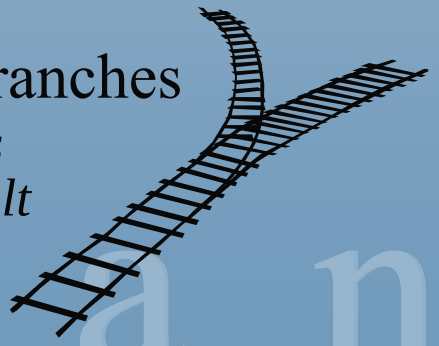
Technical Debt is Waste

Anything that makes code difficult to change

- ✓ Sloppy / Un-testable Code
  Code without a test harness is Legacy Code.

- ✓ Dependencies
  ***High cohesion** and **low coupling** are essential for testable code.*

- ✓ Unsynchronized Code Branches
  The longer two code branches remain apart, the more difficult they are to merge together.

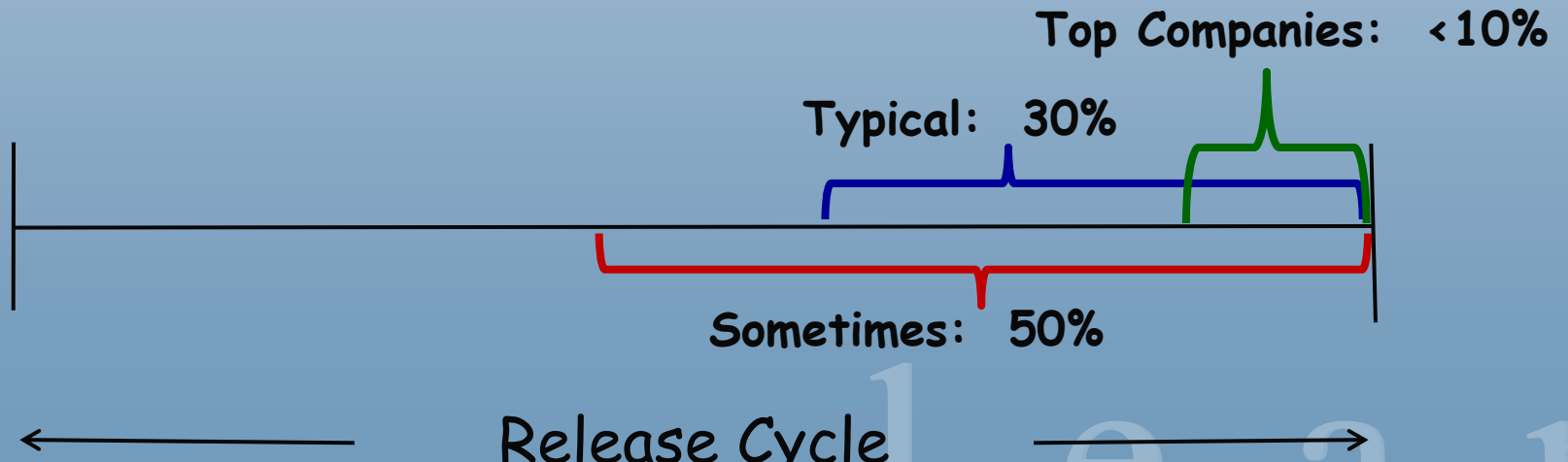


Defects are Waste

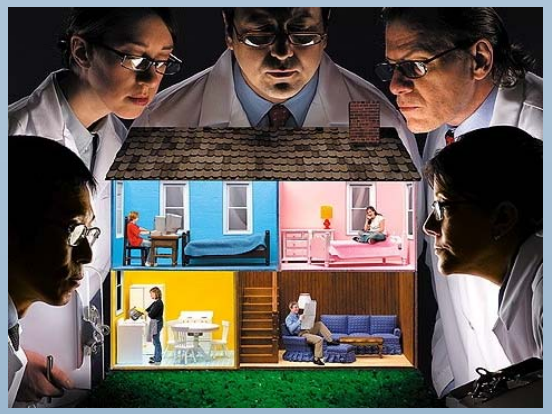
Every software development process ever invented has had the same primary goal – find and fix defects as early in the development process as possible. If you are finding defects at the end of the development process – your process is not working for you.

How good are you?

When in your release cycle do you try to freeze code and test the system?
What percent of the release cycle remains for this "hardening"?



Building the Wrong Thing is Waste



Ethnography

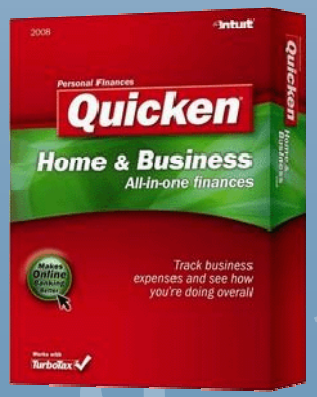
Ideation

"Go and See"
What the real customers problem is.

- ✓ Developers should talk directly with customers, ask questions, model and discuss ideas – both before and while they are developing a product.
- ✓ Intermediaries are not adequate!

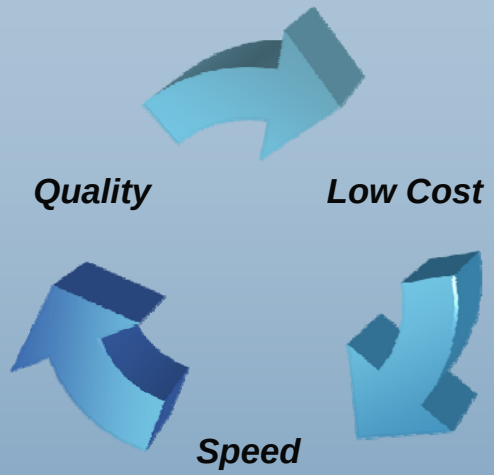
Example: Intuit's Follow-me-Home

- ✓ Whole team follows first time customers home
- ✓ Ask questions, empathize with problems
- ✓ *"It was an experience I'll never forget."*



Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



Quality by Construction

A Quality Process Builds Quality IN.

- ✓ Rather than trying to test quality in later.

Edsger Dijkstra –

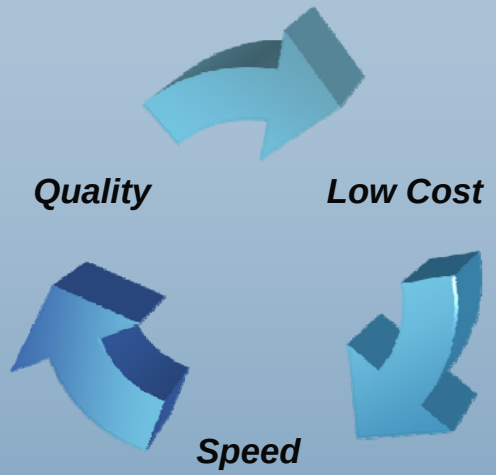
"Those who want really reliable software will discover that they must find means of avoiding the majority of bugs to start with, and as a result the programming process will become cheaper. If you want more effective programmers, you will discover that they should not waste their time debugging – they should not introduce bugs to start with."

Quality by Construction

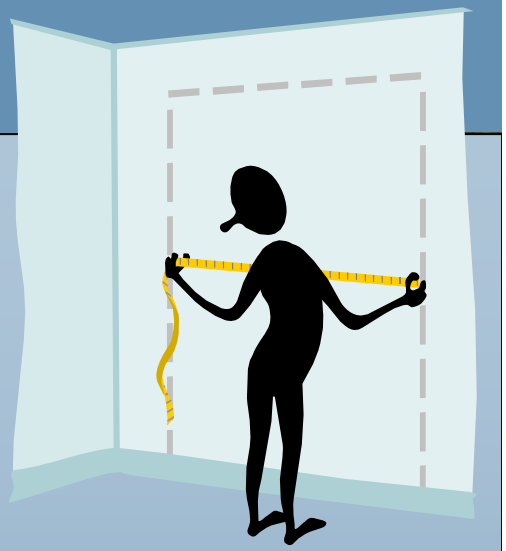
- ✓ Code that reveals its intentions
- ✓ Design/code reviews
- ✓ Immediate, automated testing
- ✓ STOP if the tests don't pass!
- ✓ Continuous, nested integration
- ✓ Escaped defect analysis & feedback

Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



Maintain Options



The Goal: ***Change-Tolerant Software***

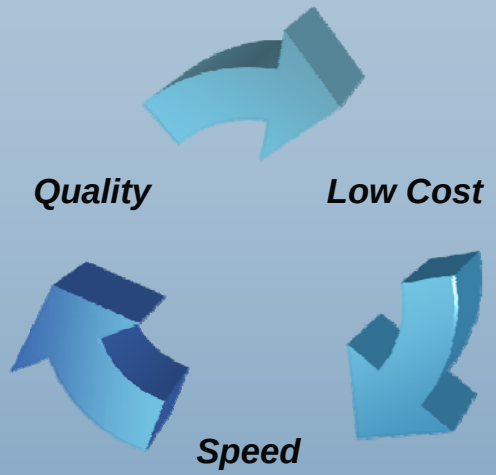
- ✓ 60-80% of all software is developed after first release to production.
- ✓ A development process that anticipates change will result in software that tolerates change.

The Strategy: ***Maintain Options***

- ✓ Make decisions ***reversible*** whenever possible.
- ✓ Make ***irreversible*** decisions as late as possible.

Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



lean

Empire State Building

September 22, 1929
Demolition started

January 22, 1930
Excavation started

March 17, 1930
Construction started

November 13, 1930
Exterior completed

May 1, 1931
Building opened

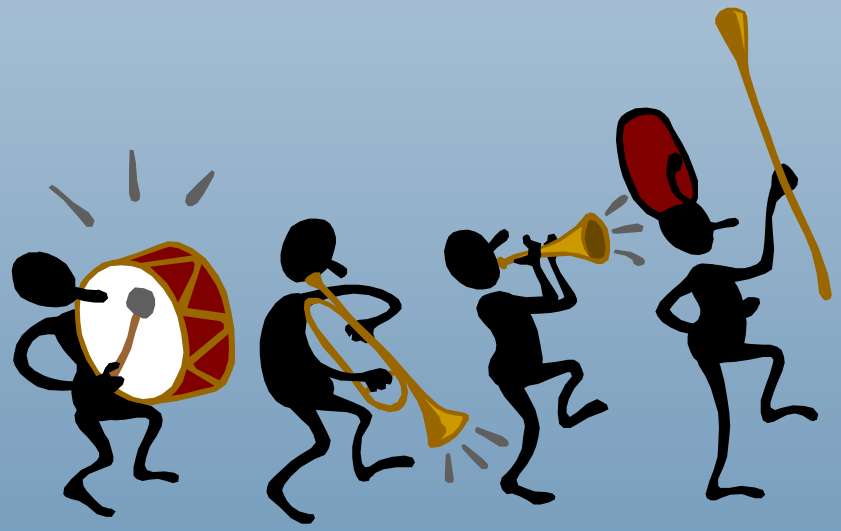
Exactly on time
18% under budget

One Year Earlier:

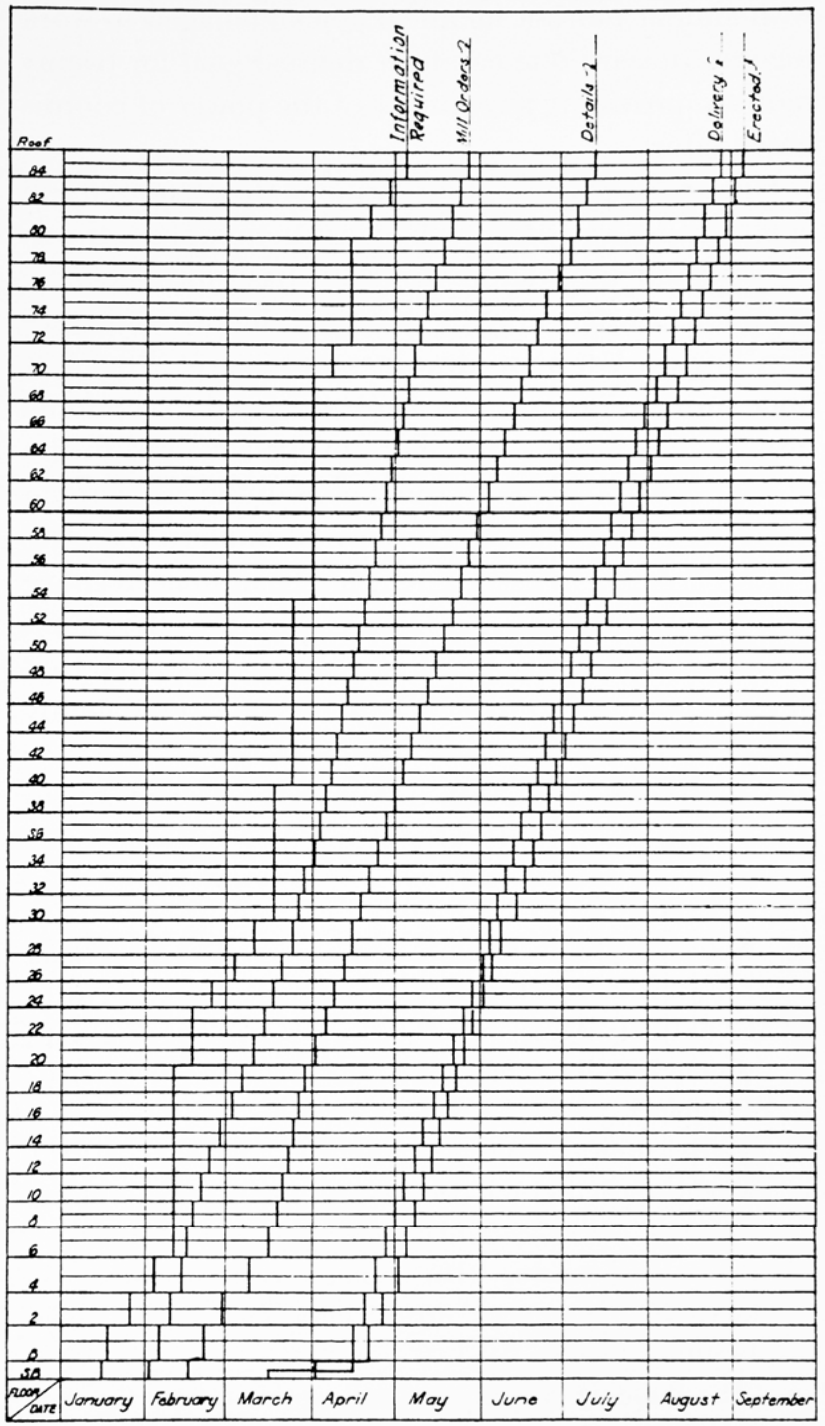
How did they do it? The key: Focus on FLOW.

Steel Schedule

We thought of the work as if it were a band marching through the building and out the top.



From: "Building the Empire State"
Builders Notebook: Edited by Carol Willis



The Four Pacemakers

1. Structural Steel Construction
   - ✓ Completed September 22, 12 days early
2. Concrete Floor Construction
   - ✓ Completed October 22, 6 days early
3. Exterior Metal Trim &Windows
   - ✓ Completed October 17, 35 days early
4. Exterior Limestone
   - ✓ Completed November 13, 17 days early

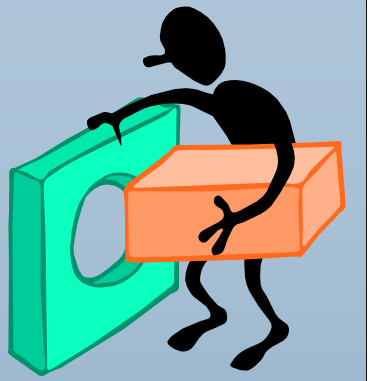
From: "Building the Empire State"
Builders Notebook: Edited by Carol Willis

Key Success Factors

1. ***Teamwork of owner, architect, and builder***
   - ✓ Eliminated design loops by consulting experts early.
2. ***Deeply Experienced Builders***
   - ✓ Fixed Price Contract!
3. ***Focus on the key constraint: Material Flow***
   - ✓ 500 trucks a day – no storage on site
4. ***Decoupling***
   - ✓ The pacemakers (and other systems) were ***designed*** to be independent
5. ***Cash Flow Thinking***
   - ✓ Every day of delay cost $10,000 ($120,000 today).
6. ***Schedule was not laid out based on the details of the building design, the building was designed based on the constraints of the situation.***
   - ✓ Two acres of land in the middle of New York City, zoning ordinances, $35,000,000 of capital, the laws of physics, and a May 1, 1931 deadline.

Lessons

Design the system to meet the constraints; do not derive constraints from the design.

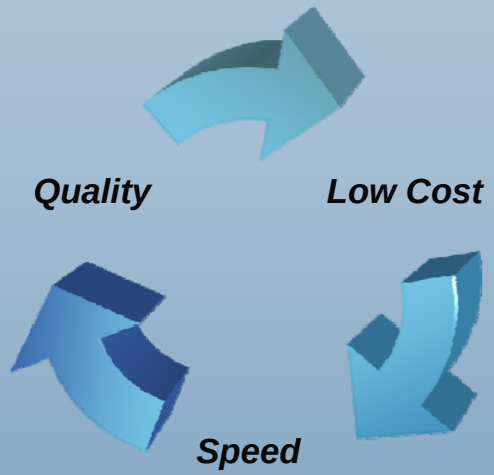


Decouple workflows; break dependencies!

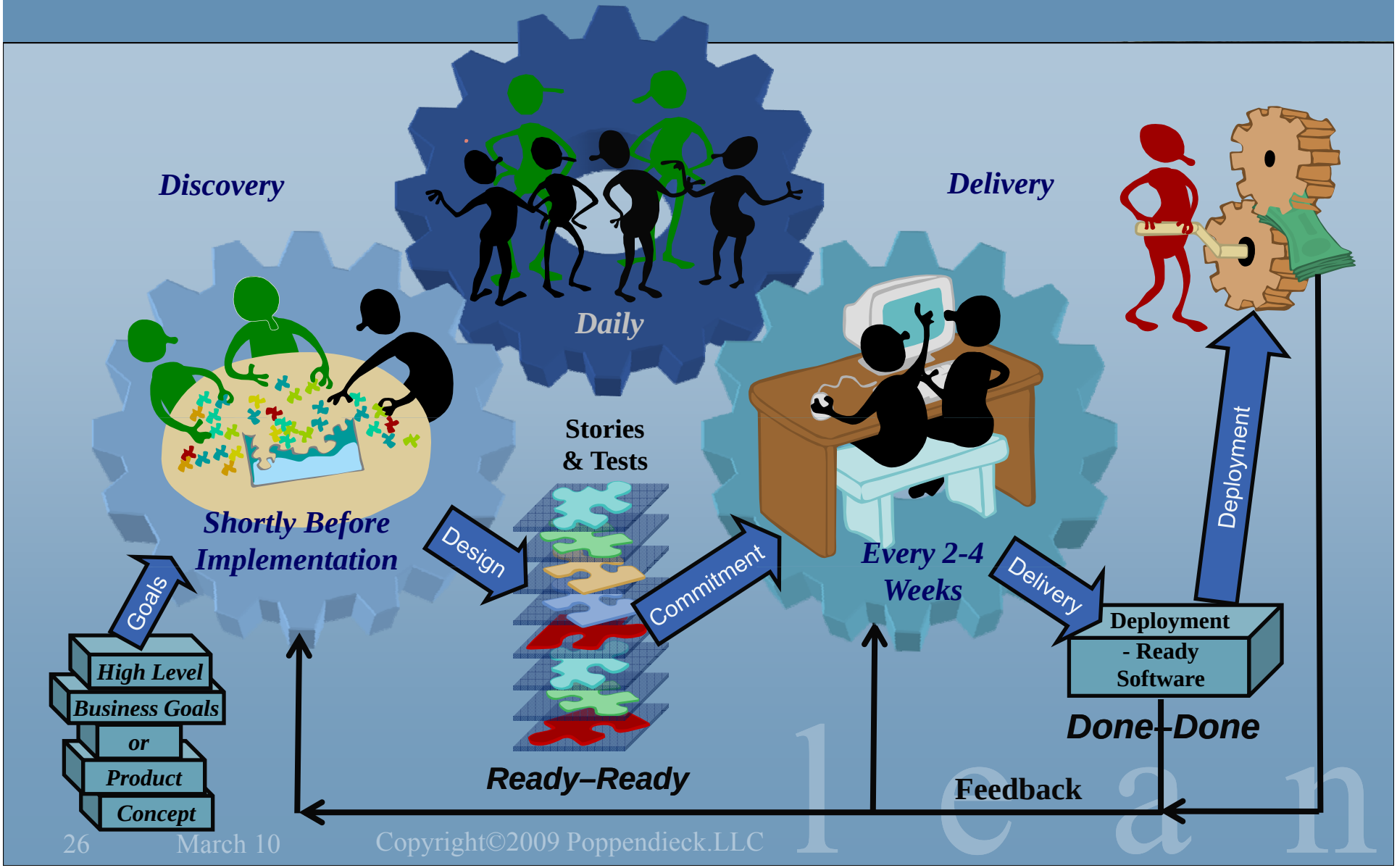
Workflows are easier to control & more predictable than schedules.

Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole

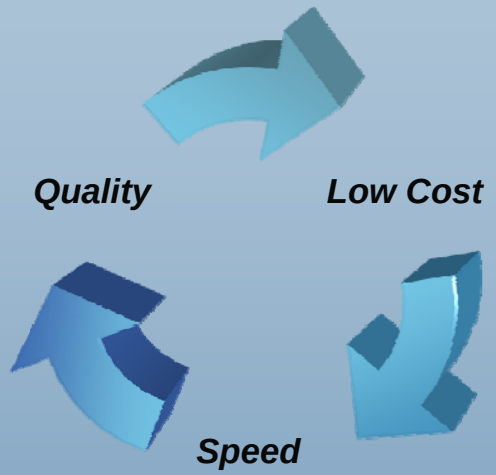


The Iterative Workflow



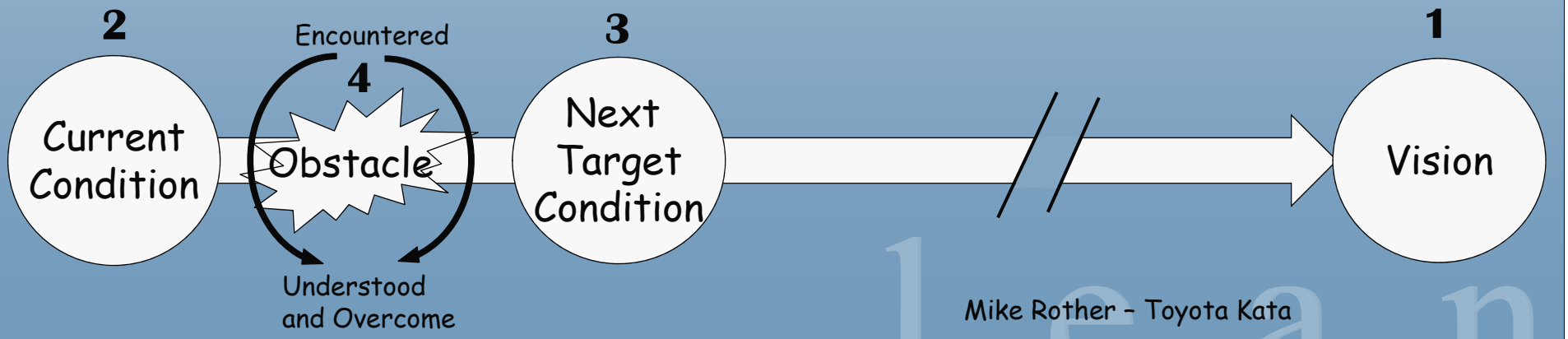
Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



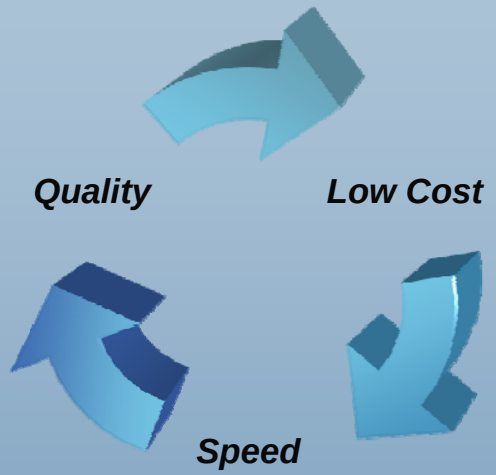
Improvement Kata

1. Visualize perfection
2. Have a first hand grasp of the situation
3. Define a target condition on the way to perfection
4. Strive to move step-by-step to the target
5. As obstacles are encountered, they are systematically understood and overcome



Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



Purpose

The Gold Standard: Tandberg – Oslo, Norway

Everyone who works there told us the same thing:

"Everything we do here is to make it easy for people to communicate."

"This is a great company. We think of programmers and salespeople as being the most important and respected positions, the "others", like the VP you met, all think of themselves – and express loudly – that they are in a supporting role." Olve Maudal, C++ Programmer

Pride

TANDBERG Codec C90

20 months from Idea to Production

Started spring 2007

1st HW prototype mid 2008

Released late 2008

55-65 people involved

- 2-3 people mechanics/design
- 4-5 people electronics/hardware
- 40-50 people software dev
- 5-6 people fpga development
- 4 people test developers
- 1 person approvals

Product Development in Tandberg

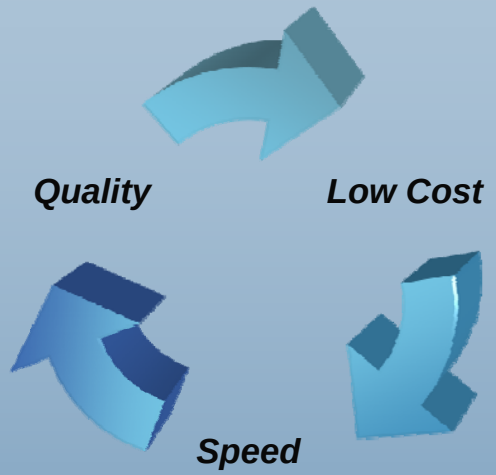
- ✓ We spend very little energy on things that are not essential
- ✓ We do not follow a plan, we do not follow procedures
- ✓ We do not write hours, we do not measure project cost
- ✓ Decisions are delayed as much as possible
- ✓ To fail is OK, therefore we deliver spectacular stuff
- ✓ Doers are very much respected in Tandberg
- ✓ We hire and keep exceptional people
- ✓ Communication is a key skill for all our engineers
- ✓ We are fast and "sloppy"
- ✓ We release early and we release often
- ✓ Little documentation gives effective communication
- ✓ Slack is embedded, and "skunk work" projects appreciated
- ✓ The company builds on trust
- ✓ Developers are treated as professionals, not as resources
- ✓ Fun and Profit

We follow principles, not processes!

Olve Maudal http://www.pvv.org/~oma/SoftwareDevelopmentAtTandberg_March2009.pdf

Principles of Lean Software Development

1. Eliminate Waste
2. Build Quality In
3. Defer Commitment
4. Deliver Fast
5. Focus on Learning
6. Improve Relentlessly
7. Respect People
8. Optimize the Whole



Appreciate the System

Zara: Fashion clothing

- ✓Design-to-Store in 2 weeks.
- ✓Twice-weekly orders.
  - Delivers globally 2 days after order
    - On hangers, priced, ready to sell
    - Shipping prices are not optimized!
- ✓Manufactures in small lots
  - Mostly at co-ops in Western Spain
    - At Western European labor rates…

RESULTS	Zara	Industry
New Items introduced / year	11,000	3,000
Items sold at full price	85%	60-70%
Unsold Items	<10%	17-20%
% sales spent on advertising	0.3%	3-4%
% sales spent on IT	0.5%	2%

Southwest Airlines

1973: Had to sell 1 of 4 airplanes

Decided not to reduce its routes, but to reduce its turnover time instead.

Why? Because it was competing against driving, not against other airlines.

Discovered it could run the same routes using ¾ of the assets if it focused on:

Planes in the Air

NOT Filling Seats.

America's largest, most profitable, and most popular and admired airline.

"The secret to success is to take good care of your employees, then they will take good care of your customers, and satisfied customers lead to a successful business."

l e a n
software development

Thank You!

More Information: www.poppendieck.com

mary@poppendieck.com Mary Poppendieck www.poppendieck.com