What Does It Really Mean to be Agile?

Siew Kok Ewe, CST
kok.ewe.siew@gmail.com

Agile Gurgaon 2016

Hello!

- Siew Kok Ewe ("KE")
- Certified Scrum Trainer®
- Lives in Penang

©OnlyPenang

© Simone Tramonte / Barcroft Media

PassportChop.com

AMEED'S
TEH TARIK
SINCE 1978
CAP BOXING

Biscuits
89 槟城淡汶饼
土產食品
加料粿條湯
Special KOAY TEOW TH'NG

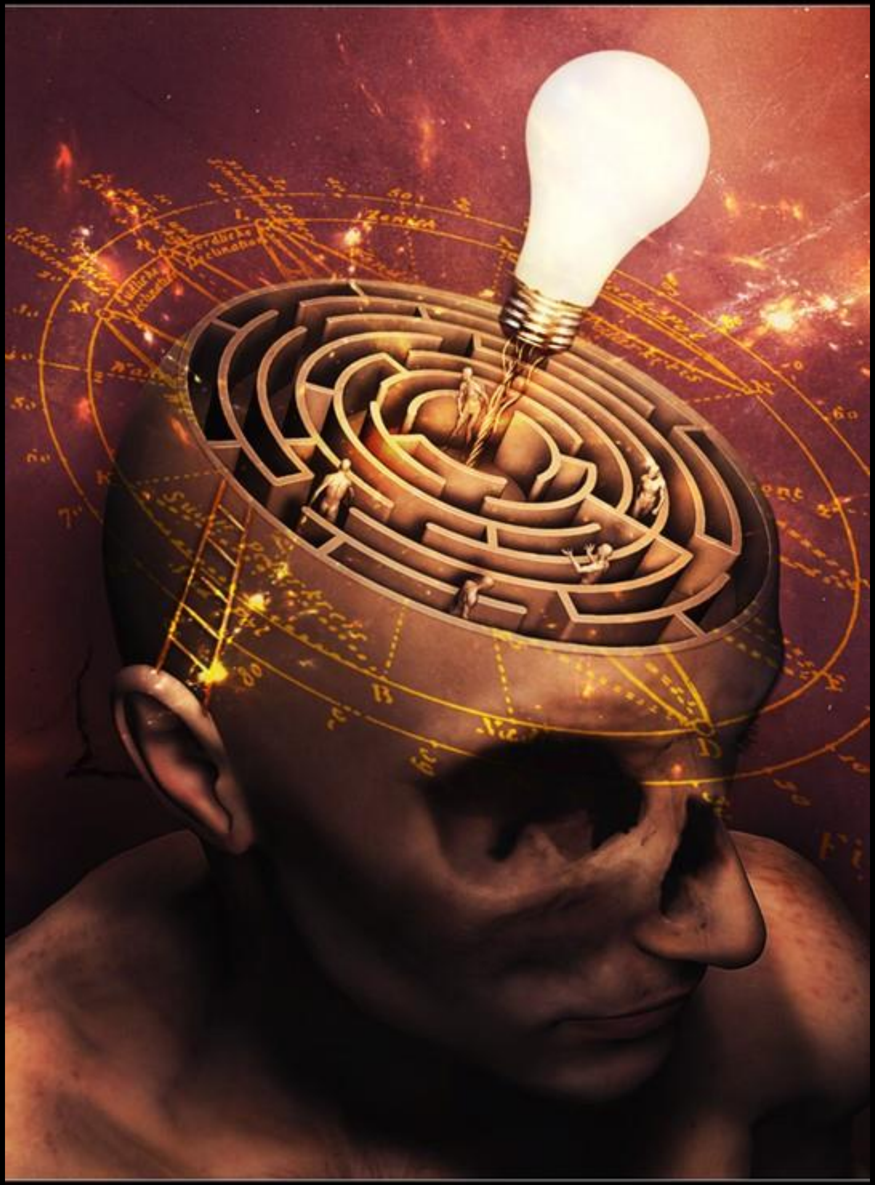
Product Manager & Developers
Grow & Work in a crossfunctional TEAM
self organizing team
HIGHEST PRIORITY
CUSTOMER SATISFACTION
& high process product quality
visual
AGILE
MANIFESTO
© Pritam Sen
shorter development iteration
continuos inspection
continuos delivery
BUILD PRODUCT around
sustainable development pace
measure development progress
working documentation
Identify Understand and Harness the forces of change
Keep it simple
CHANGING REQUIREMENTS ARE welcome
effective FACE to FACE communication
motivated individuals

What is Agile Really?

Traditional vs. Agile

Getting There

What is Agile, Really?



Product Development work is

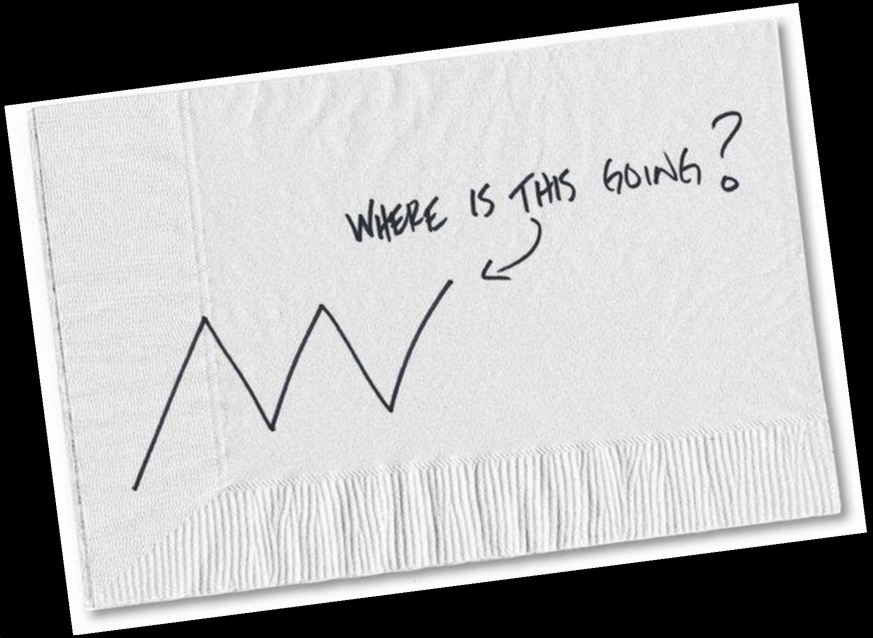
complex

Customers

Change their mind

Team

Doesn't have full knowledge



What are they up to?

Competitors

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

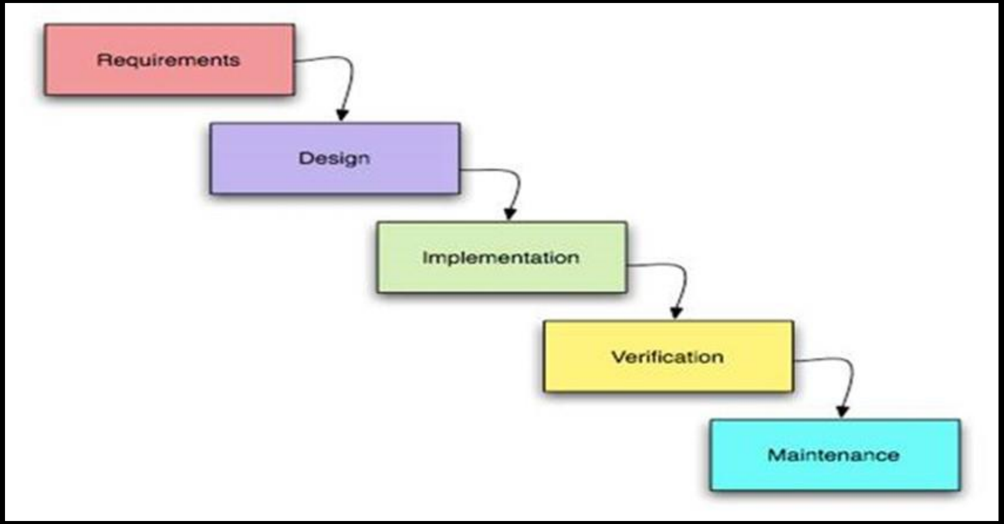
Traditional vs. Agile

Traditional vs. Agile

Process, Structure, Management

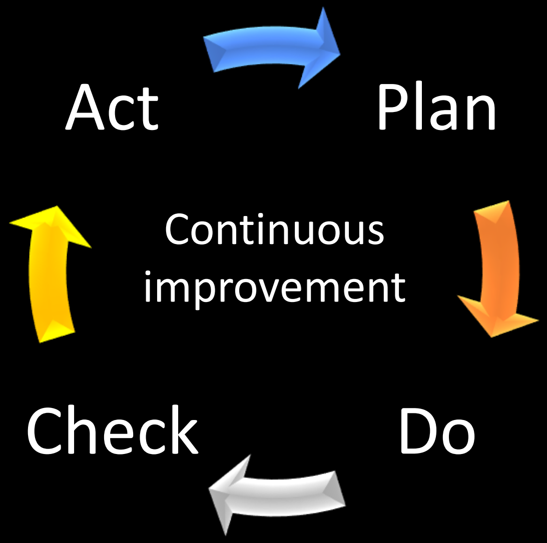
Process

Traditional



Defined Process
Predict & Control
Sequential activities

Agile



Act

Plan

Continuous improvement

Check

Do

Empirical Process
Inspect & Adapt
Incremental & Iterative

Process

#1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.

#2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.

#3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.

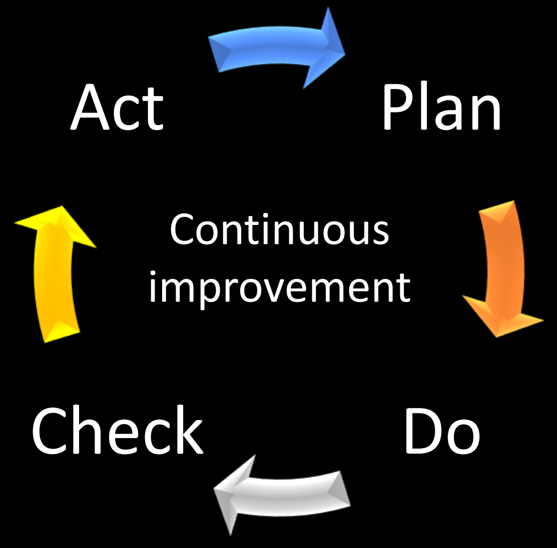
#7. Working software is the primary measure of progress.

#9. Continuous attention to technical excellence and good design enhances agility.

#10. Simplicity – the art of maximizing the amount of work not done – is essential.

#12. At regular intervals, the team reflects on how to become more effective, then tunes and adjust its behaviour accordingly.

Agile



Act

Plan

Continuous improvement

Check

Do

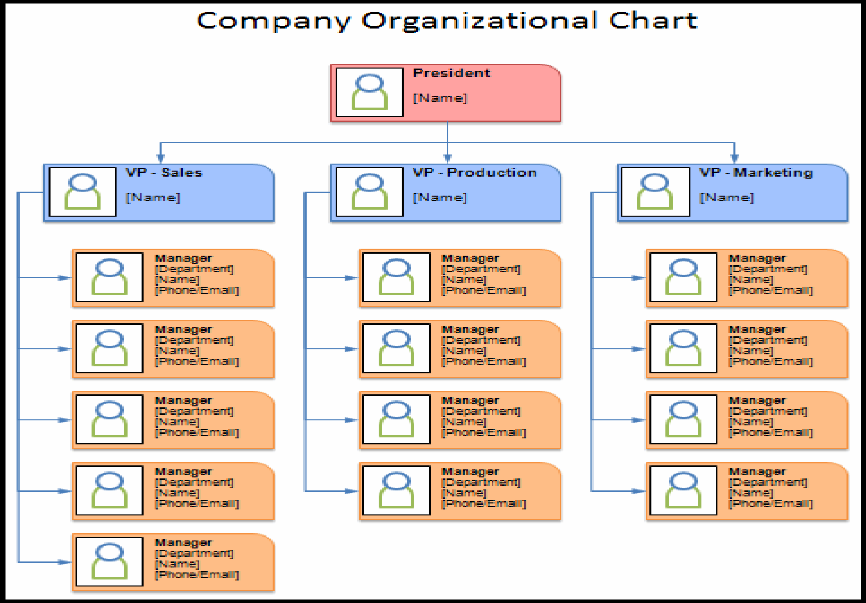
Empirical Process

Inspect & Adapt

Incremental & Iterative

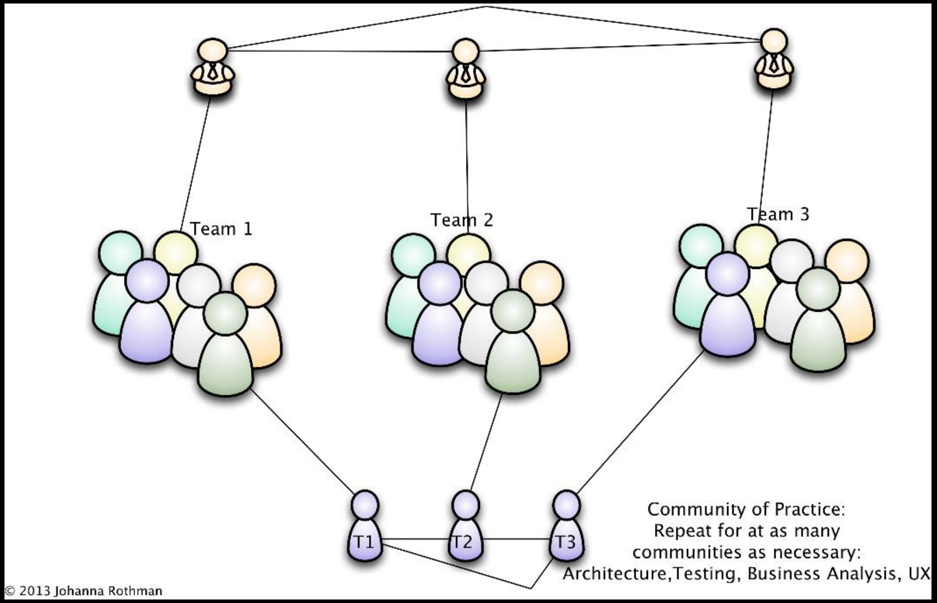
Structure

Traditional



Hierarchy of Functional Departments

Agile



Community of Practice:
Repeat for at as many
communities as necessary:
Architecture,Testing, Business Analysis, UX

© 2013 Johanna Rothman

Network of Cross-functional Teams

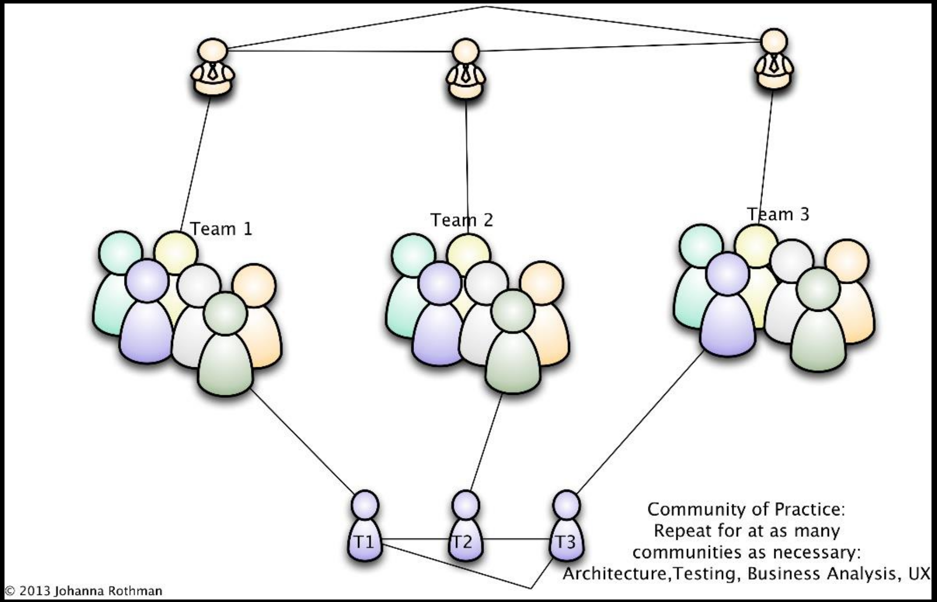
Structure

Agile

#4. Business people and developers must work together daily throughout the project.

#5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

#6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.



Community of Practice:
Repeat for at as many
communities as necessary:
Architecture,Testing, Business Analysis, UX

© 2013 Johanna Rothman

Network of Cross-functional Teams

Management

Traditional

Command and Control Management

Agile

Self-Management

Management

Agile

#8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

#11. The best architectures, requirements, and designs emerge from self-organizing teams.

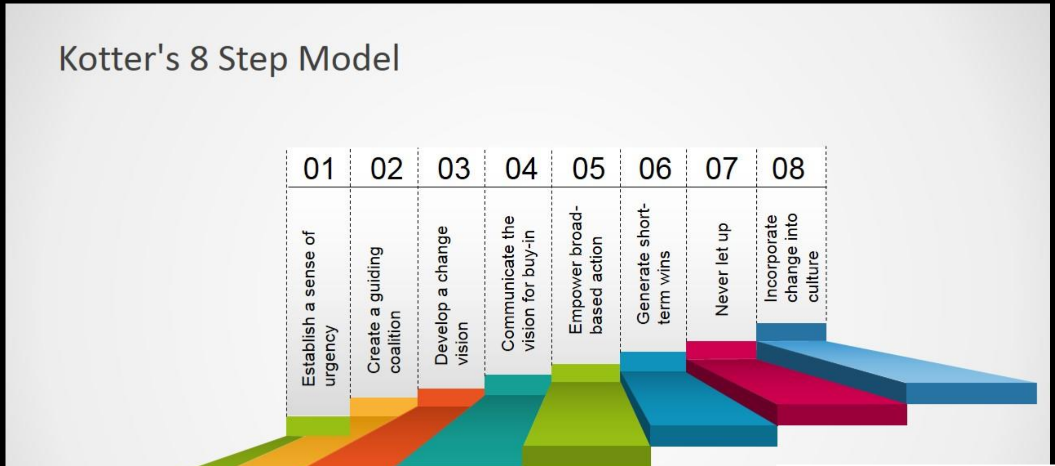
Self-Management

Getting There

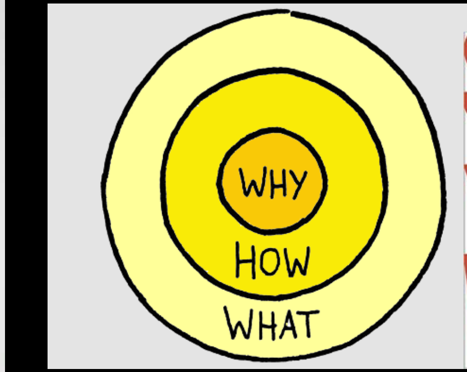
Traditional to Agile

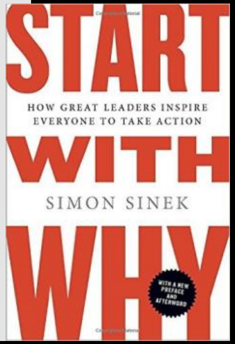
Change

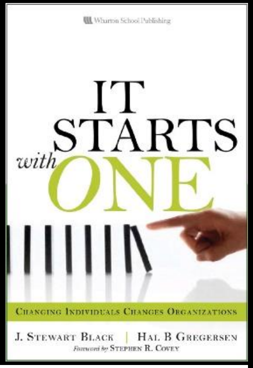
Change Models

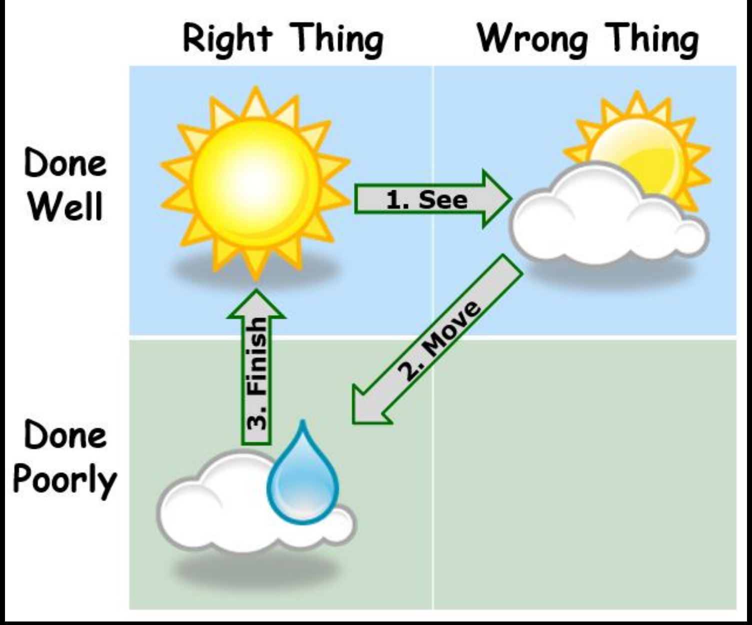

Kotter's 8 Step Model

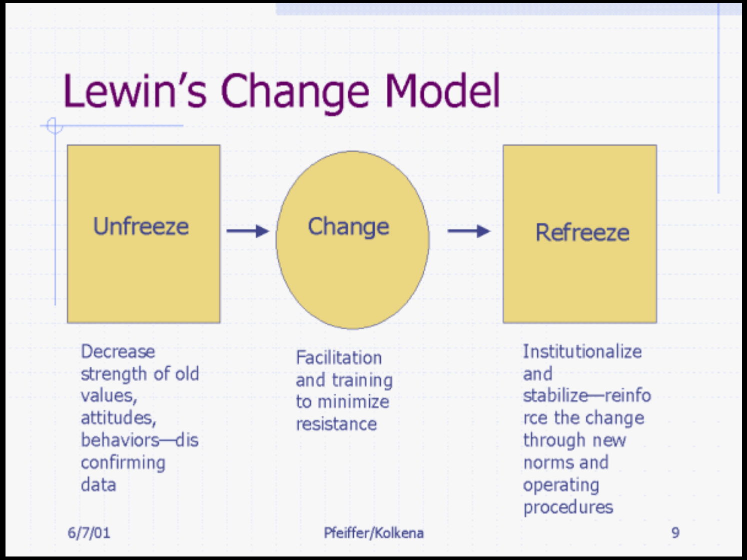
01
Establish a sense of urgency
02
Create a guiding coalition
03
Develop a change vision
04
Communicate the vision for buy-in
05
Empower broad-based action
06
Generate short-term wins
07
Never let up
08
Incorporate change into culture




START
HOW GREAT LEADERS INSPIRE EVERYONE TO TAKE ACTION
WITH
SIMON SINEK
WHY
WITH A NEW PREFACE AND AFTERWORD


Wharton School Publishing
IT STARTS with ONE
Changing Individuals Changes Organizations
J. Stewart Black | Hal B Gregersen
Foreword by Stephen R. Covey




Lewin's Change Model

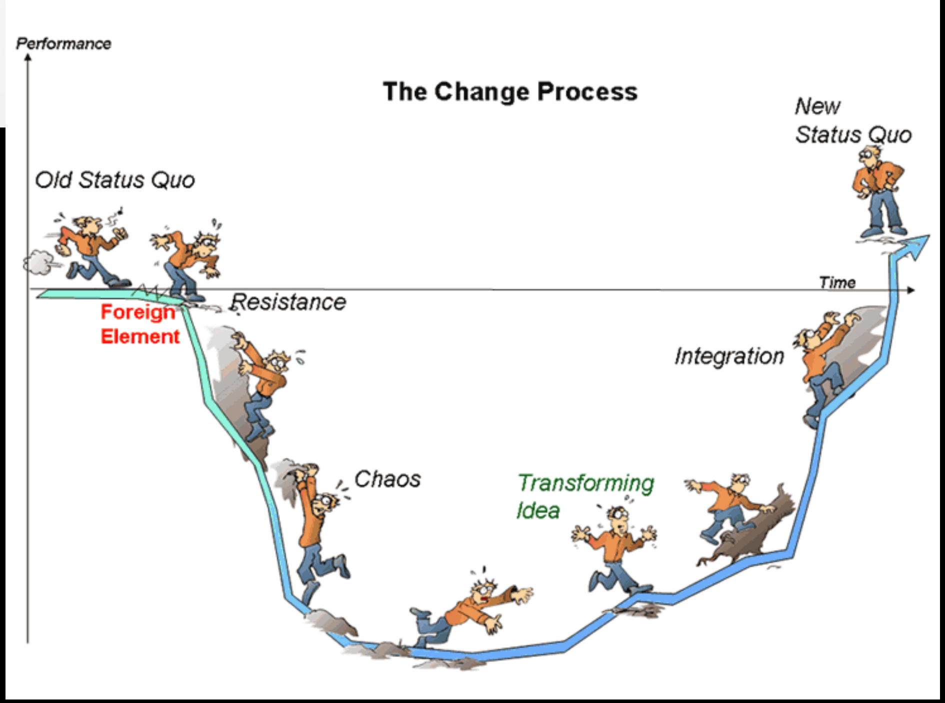
Unfreeze → Change → Refreeze

Unfreeze: Decrease strength of old values, attitudes, behaviors—dis confirming data

Change: Facilitation and training to minimize resistance

Refreeze: Institutionalize and stabilize—reinfo rce the change through new norms and operating procedures

6/7/01 Pfeiffer/Kolkena 9


The Change Process

Traditional to Agile

Change

Where They Were

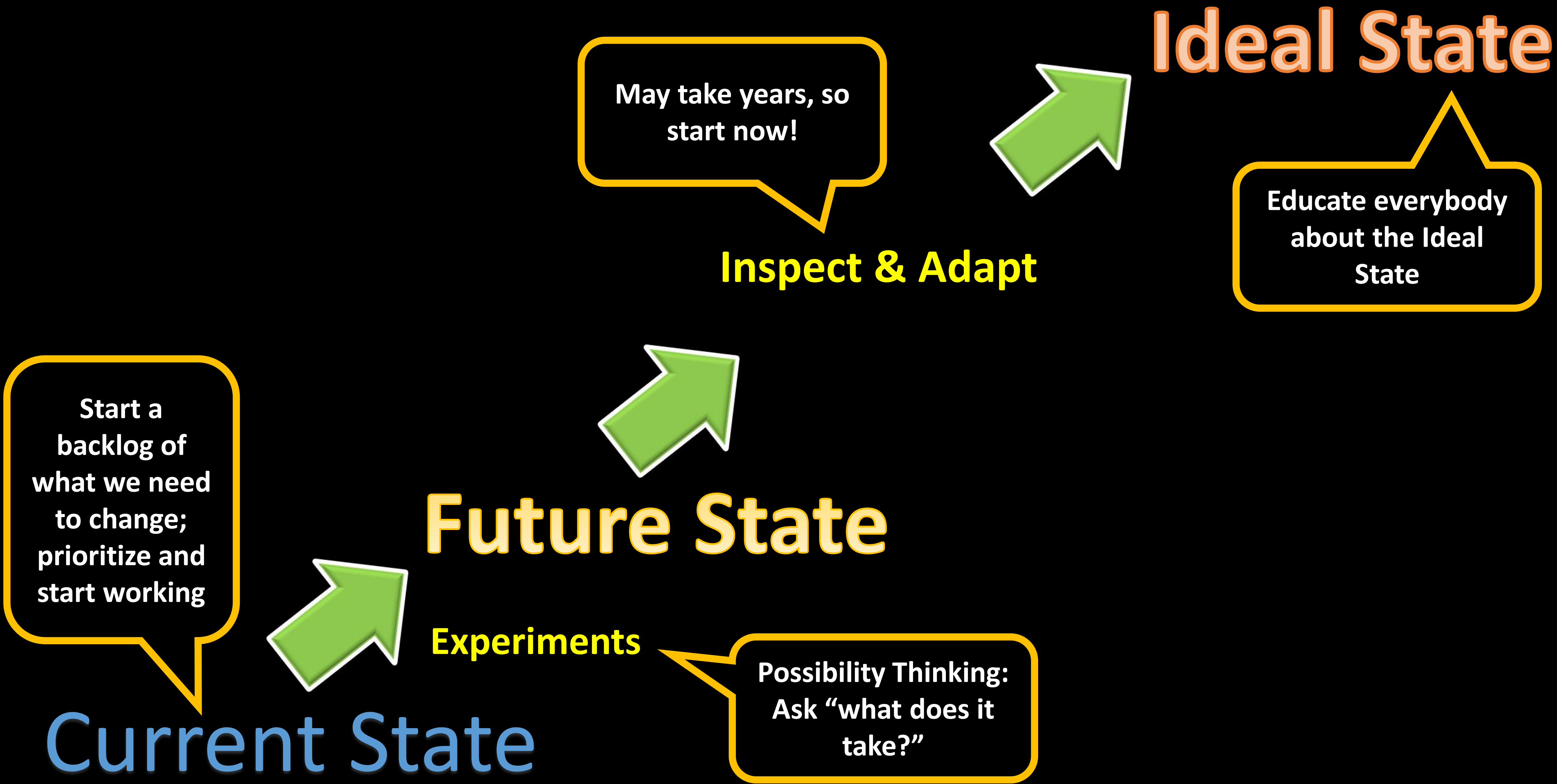
- Poor Quality
- Missing Schedules
- High Attrition
- High Pressure
- Challenging Roadmap

What They Did

- New Software Process
- New Problem Solving Process
- New Tools
- Leadership & Coaching
- Learning Organization

What They Achieved

- Higher Productivity
- Higher Morale
- Higher Quality
- Better Return on Capital
- Lower Product Cost



Current State

Start a backlog of what we need to change; prioritize and start working

Experiments

Possibility Thinking: Ask “what does it take?”

Future State

Inspect & Adapt

May take years, so start now!

Ideal State

Educate everybody about the Ideal State

Ideal State

Educate everybody about the Ideal State

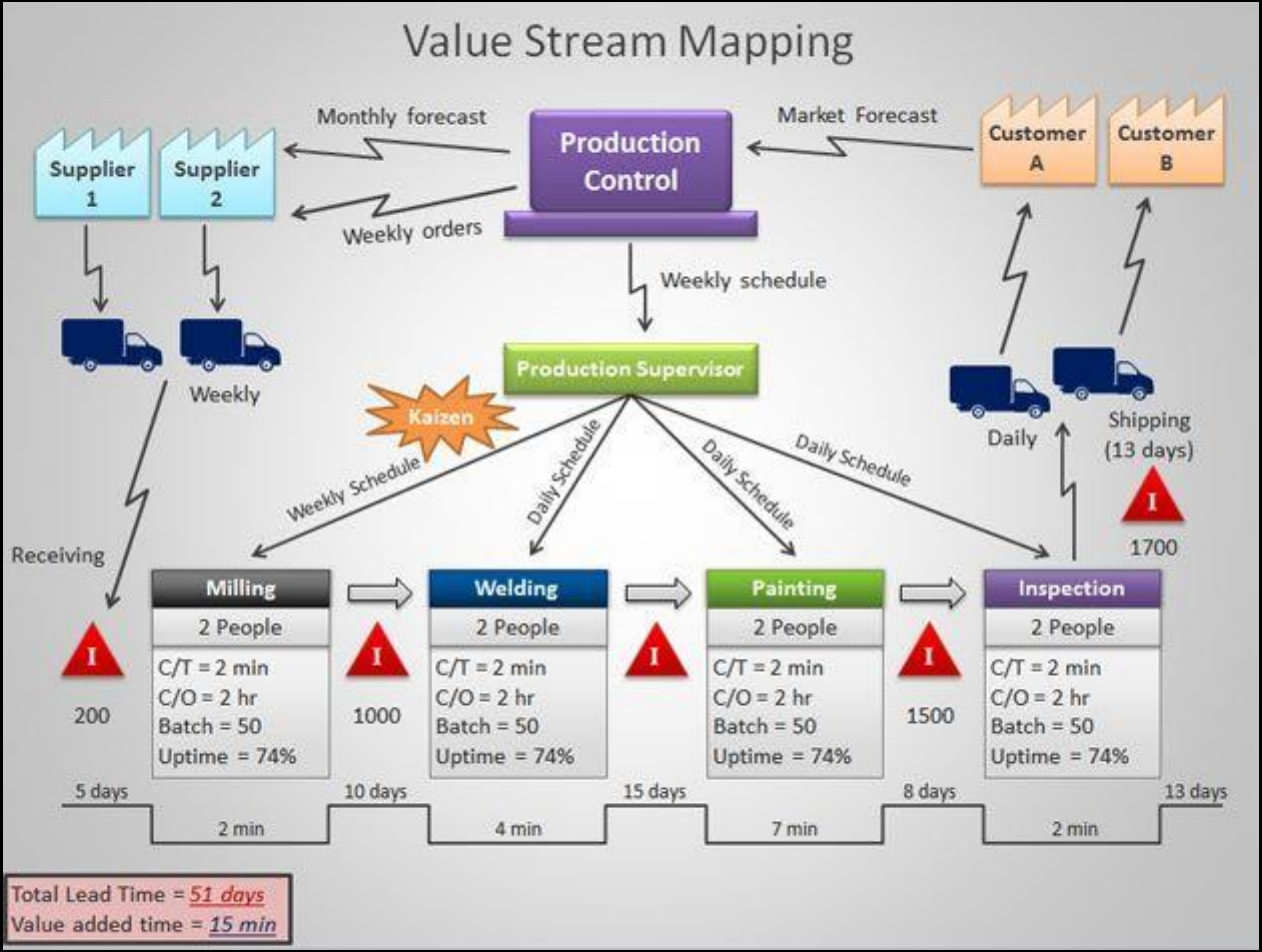
- Educate, educate, educate
  - Help everyone understand what the Ideal State is
  - Training, coaching, sharing, influencing
- For example, “Ideally, we want to be able to ship working software every day.”
- “That will never work in the *real* world…”
- We may never reach the Ideal State, but getting 20% there is better than getting 0% there.

- Where are we currently?
- What’s the gap between Current and Ideal States?
- Collect input from everybody about what to change.
- Establish a prioritized change backlog and start working.
- Understand the Current State
  - Use direct observation
  - Measure them
  - Visualize them e.g. use Value Stream Mapping
- For example, “Right now, it takes us 6-12 weeks to make a change in our software and release it to production.”

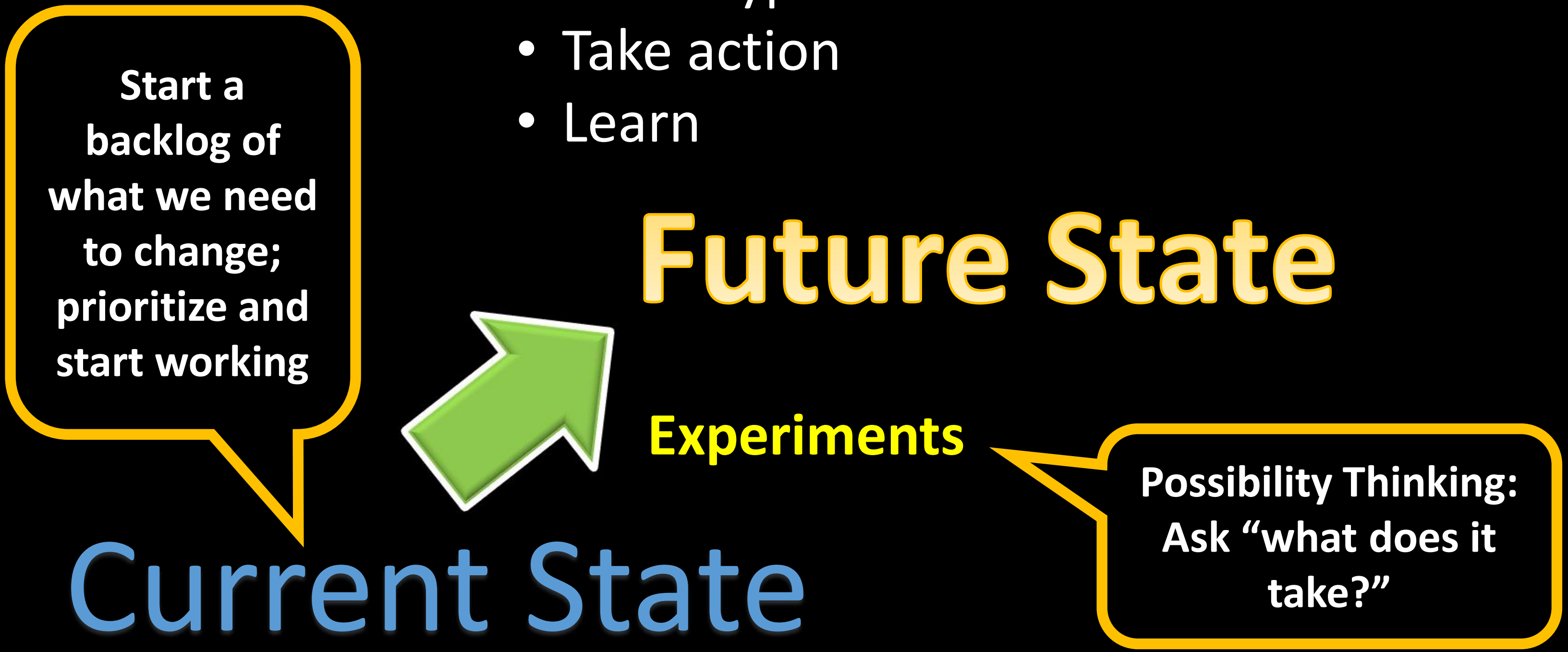
Start a backlog of what we need to change; prioritize and start working

Current State

Value Stream Mapping



- Establish a Future State after understanding Current State
  - Understand the gap between
- For example, “What does it take to make a change and release it to production in 4 weeks consistently?”
- Do experiments
  - Analyse root causes
  - Form hypothesis
  - Take action
  - Learn



Start a backlog of what we need to change; prioritize and start working

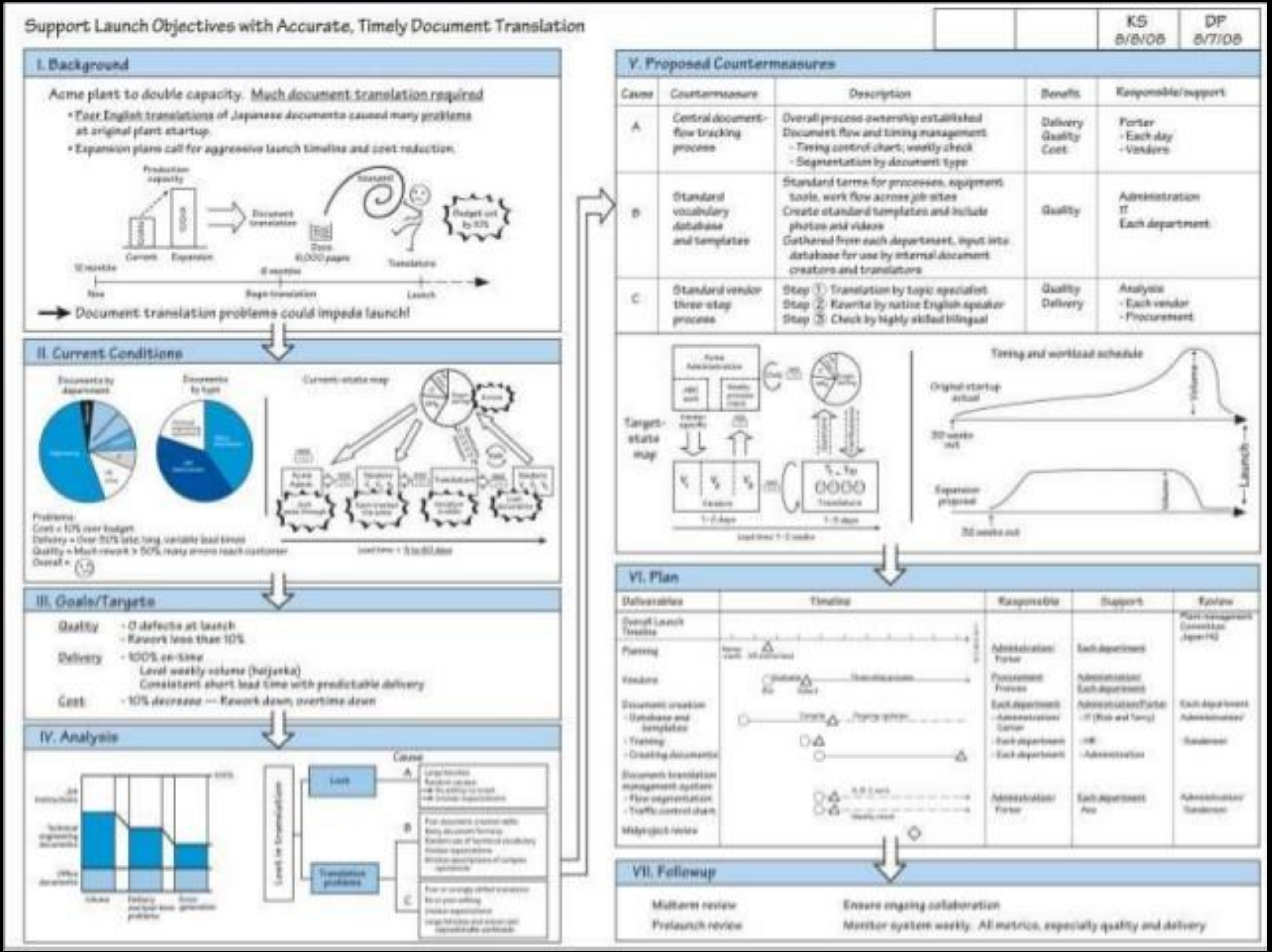
Future State

Experiments

Possibility Thinking: Ask “what does it take?”

Current State

A3 Problem Solving



Support Launch Objectives with Accurate, Timely Document Translation

		KS 8/8/08	DP 8/7/08

I. Background

Acme plant to double capacity. Much document translation required

- Poor English translations of Japanese documents caused many problems at original plant startup.
- Expansion plans call for aggressive launch timeline and cost reduction.

Production capacity
Current
Expansion
Document translation
Docs 6,000 pages
Translators
Budget cut by 10%
12 months
Now
6 months
Begin translation
Launch

→ Document translation problems could impede launch!

II. Current Conditions

Problems:
Cost = 10% over budget
Delivery = Over 50% late; long, variable lead times
Quality = Much rework > 50%; many errors reach customer
Overall =

Lead time = 5 to 60 days

III. Goals/Targets

Quality - 0 defects at launch
- Rework less than 10%

Delivery - 100% on-time
Level weekly volume (heijunka)
Consistent short lead time with predictable delivery

Cost - 10% decrease — Rework down, overtime down

IV. Analysis

V. Proposed Countermeasures

Cause	Countermeasure	Description	Benefit	Responsible/support
A	Central document-flow tracking process	Overall process ownership established Document flow and timing management - Timing control chart; weekly check - Segmentation by document type	Delivery Quality Cost	Porter - Each day - Vendors
B	Standard vocabulary database and templates	Standard terms for processes, equipment, tools, work flow across job sites Create standard templates and include photos and videos Gathered from each department, input into database for use by internal document creators and translators	Quality	Administration IT Each department
C	Standard vendor three-step process	Step ① Translation by topic specialist Step ② Rewrite by native English speaker Step ③ Check by highly skilled bilingual	Quality Delivery	Analysis - Each vendor - Procurement

VI. Plan

Deliverables	Timeline	Responsible	Support	Review
Overall Launch Timeline				Plant management Committee Japan HQ
Planning		Administration/ Porter	Each department	
Vendors		Procurement/ Friesen	Administration/ Each department	
Document creation - Database and templates - Training - Creating documents		Each department - Administration/ Carter - Each department - Each department	Administration/Porter - IT - HR - Administration	Each department Administration/ Sanderson
Document translation management system - Flow segmentation - Traffic control chart		Administration/ Porter	Each department Ako	Administration/ Sanderson
Midproject review				

VII. Followup

Midterm review — Ensure ongoing collaboration

Prelaunch review — Monitor system weekly. All metrics, especially quality and delivery

This process is built into the Scrum framework, so do it!

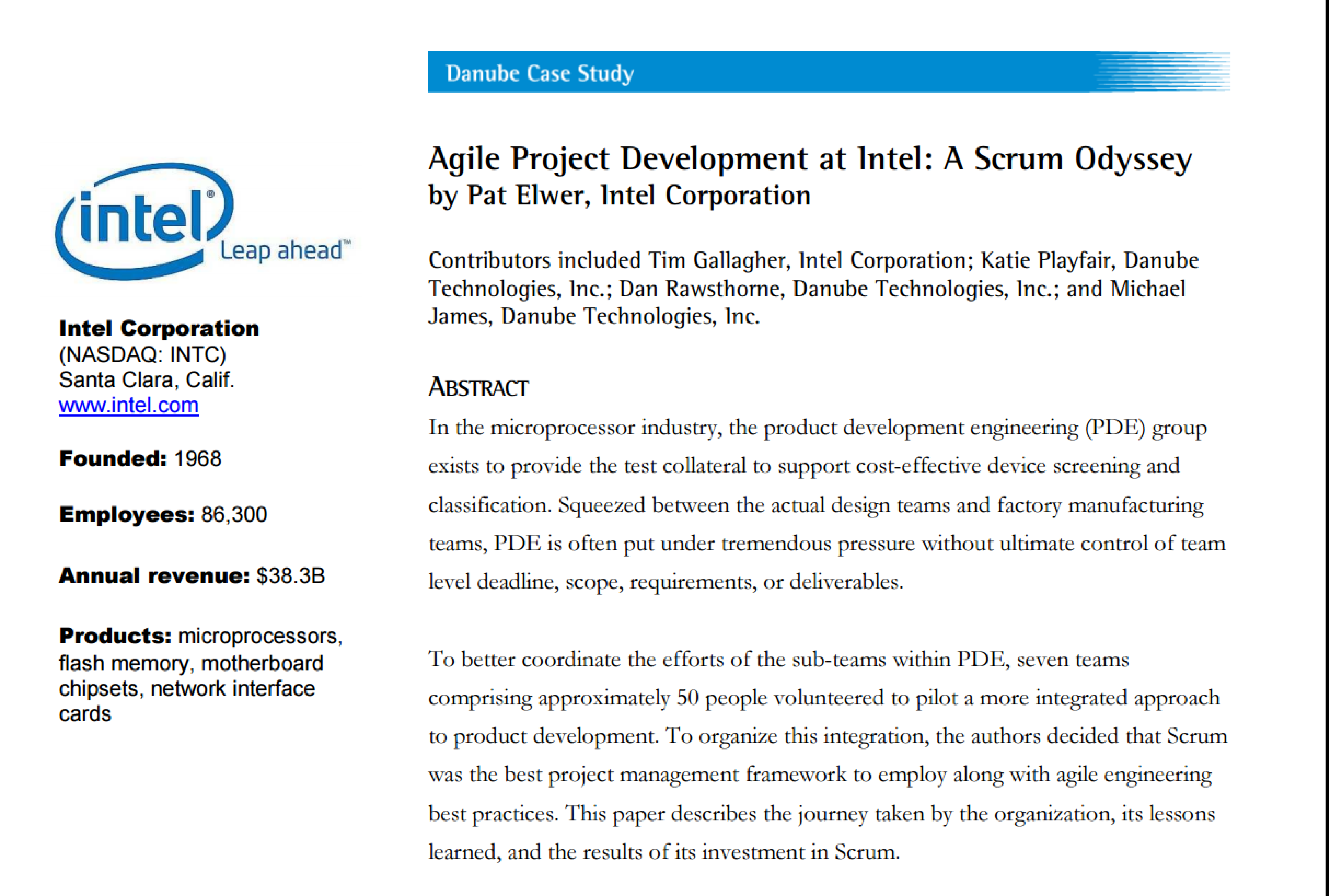
Ideal State

May take years, so start now!

Inspect & Adapt

Educate everybody about the Ideal State

- Learn from the experiments
  - Success and failures
- Everybody's responsibility to learn and continuously improve
  - Not just leaders and managers
- Employ ScrumMasters or Agile Coaches



Danube Case Study

Agile Project Development at Intel: A Scrum Odyssey

by Pat Elwer, Intel Corporation

Contributors included Tim Gallagher, Intel Corporation; Katie Playfair, Danube Technologies, Inc.; Dan Rawsthorne, Danube Technologies, Inc.; and Michael James, Danube Technologies, Inc.

ABSTRACT

In the microprocessor industry, the product development engineering (PDE) group exists to provide the test collateral to support cost-effective device screening and classification. Squeezed between the actual design teams and factory manufacturing teams, PDE is often put under tremendous pressure without ultimate control of team level deadline, scope, requirements, or deliverables.

To better coordinate the efforts of the sub-teams within PDE, seven teams comprising approximately 50 people volunteered to pilot a more integrated approach to product development. To organize this integration, the authors decided that Scrum was the best project management framework to employ along with agile engineering best practices. This paper describes the journey taken by the organization, its lessons learned, and the results of its investment in Scrum.

Intel Corporation
(NASDAQ: INTC)
Santa Clara, Calif.
www.intel.com

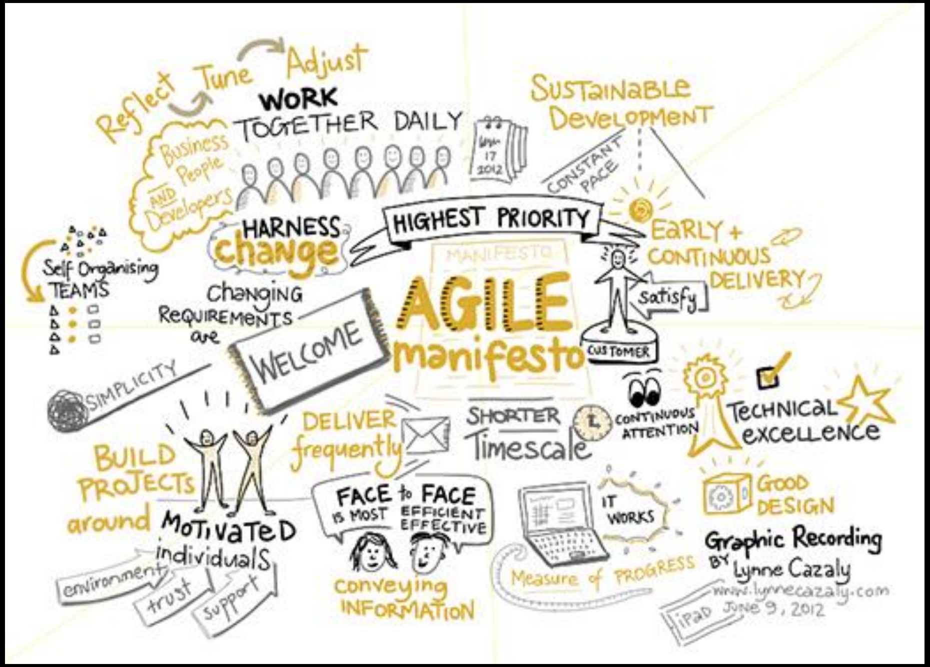
Founded: 1968

Employees: 86,300

Annual revenue: $38.3B

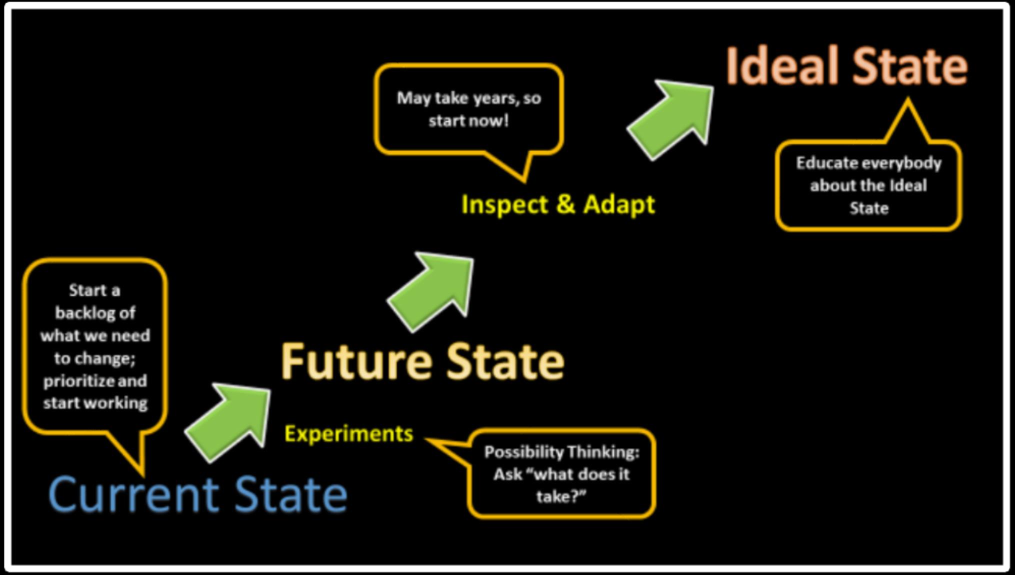
Products: microprocessors, flash memory, motherboard chipsets, network interface cards

Traditional vs Agile



Agile

Getting There



Ideal State

Educate everybody about the Ideal State

May take years, so start now!

Inspect & Adapt

Future State

Experiments

Possibility Thinking: Ask "what does it take?"

Start a backlog of what we need to change; prioritize and start working

Current State

Thank you...