SAFETY NOT GUARANTEED

How great teams break the rules to create successful products

Jeff Patton
Chief troublemaker, Comakers LLC
jeff@comakewith.us, twitter: @jeffpatton

In software development,

we fail a lot

(cc) BY-NC-SA

September 2011: UK government scraps the £12 billion National program for IT in the NHS

The U.K.'s Major Projects Authority concluded the project was "not fit to provide the modern IT services that the NHS needs."

New York City's CityTime project originally had a $63 million budget, but costs since skyrocketed astonishingly, with total estimates reportedly reaching $760 million.

November 2012, the United States Air Force canceled a six-year-old modernization effort that had eaten up more than $1 billion.

The Air Force canceled after realizing that it would cost another $1 billion just to achieve one-quarter of the capabilities originally planned.

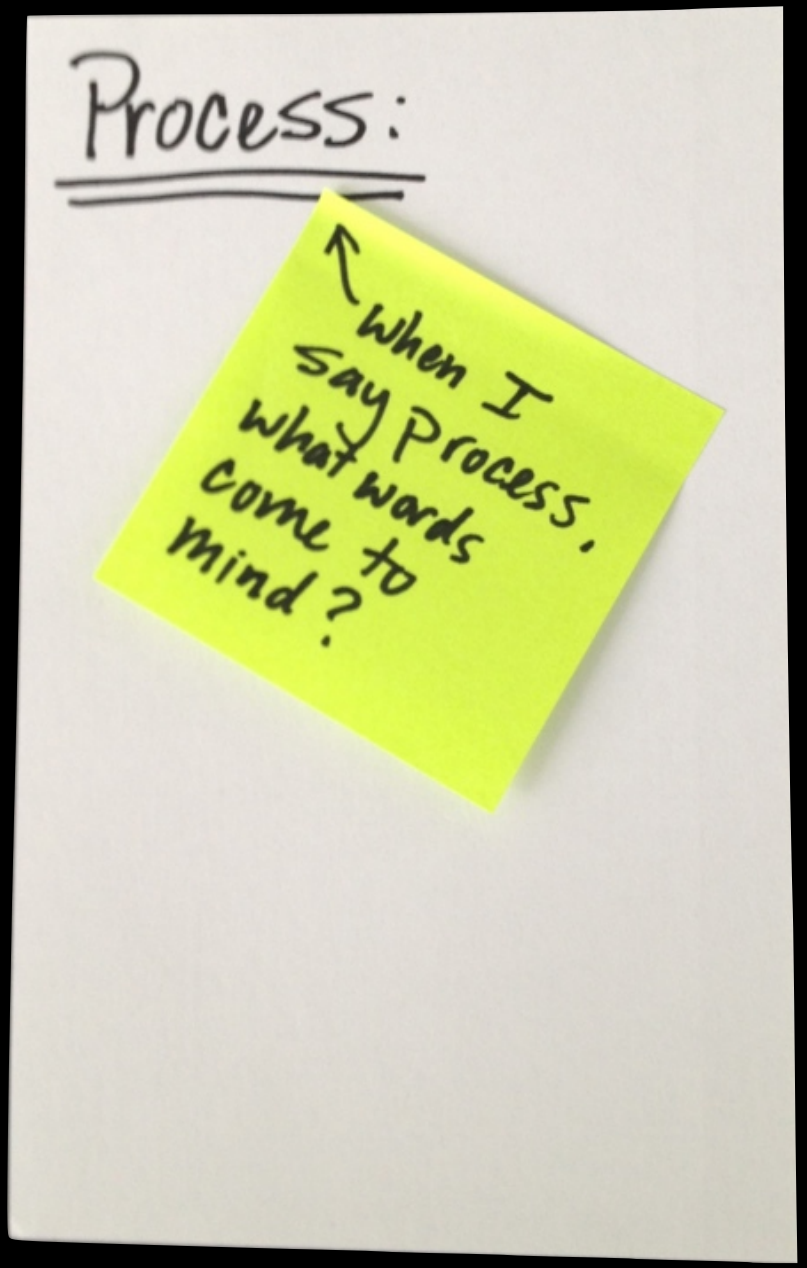
It didn't take long to find these numbers...

(cc) BY-NC-SA

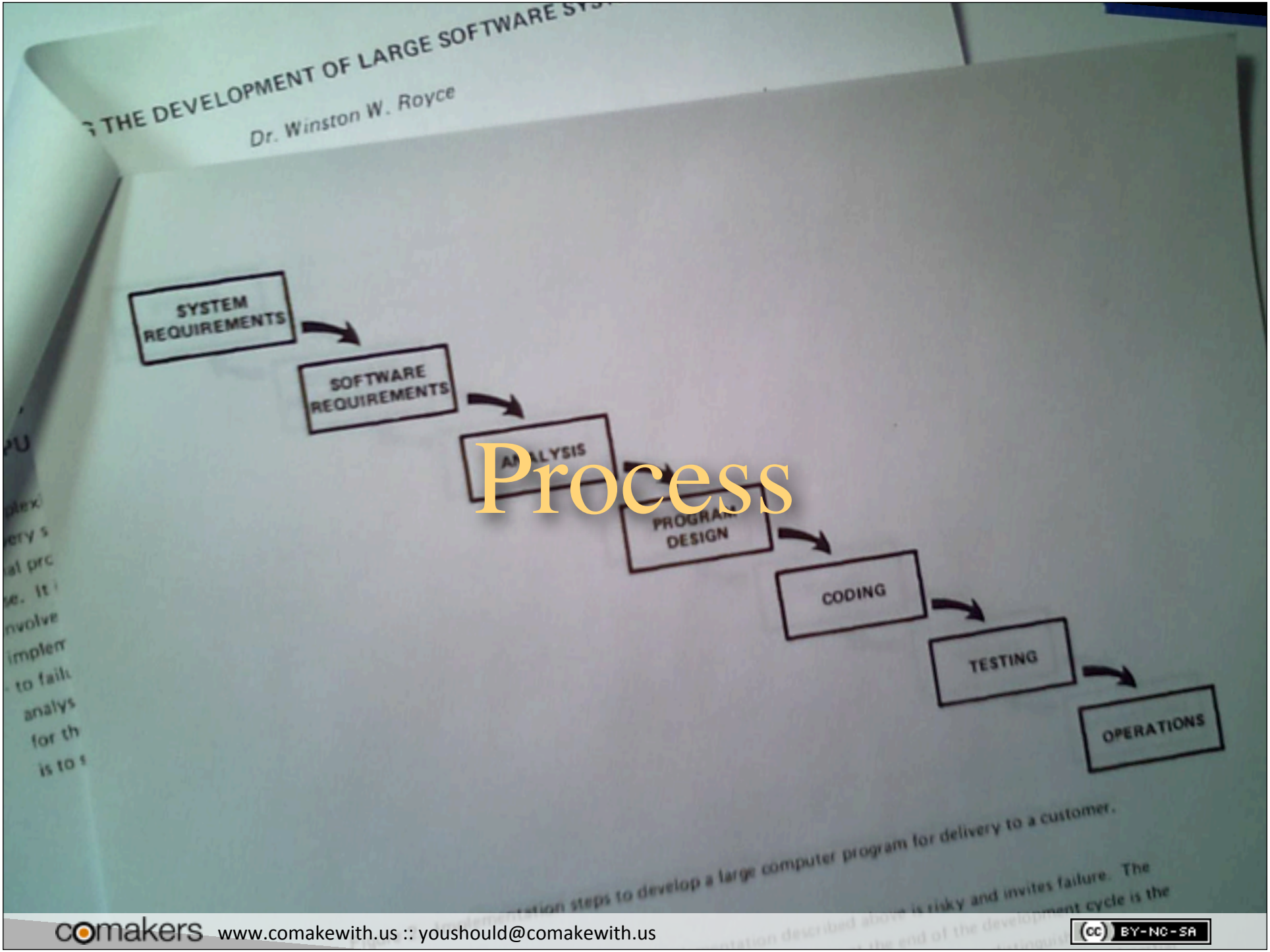
This is a talk about process, and what you can and can't do about all this

(cc) BY-NC-SA

What does process mean to you?

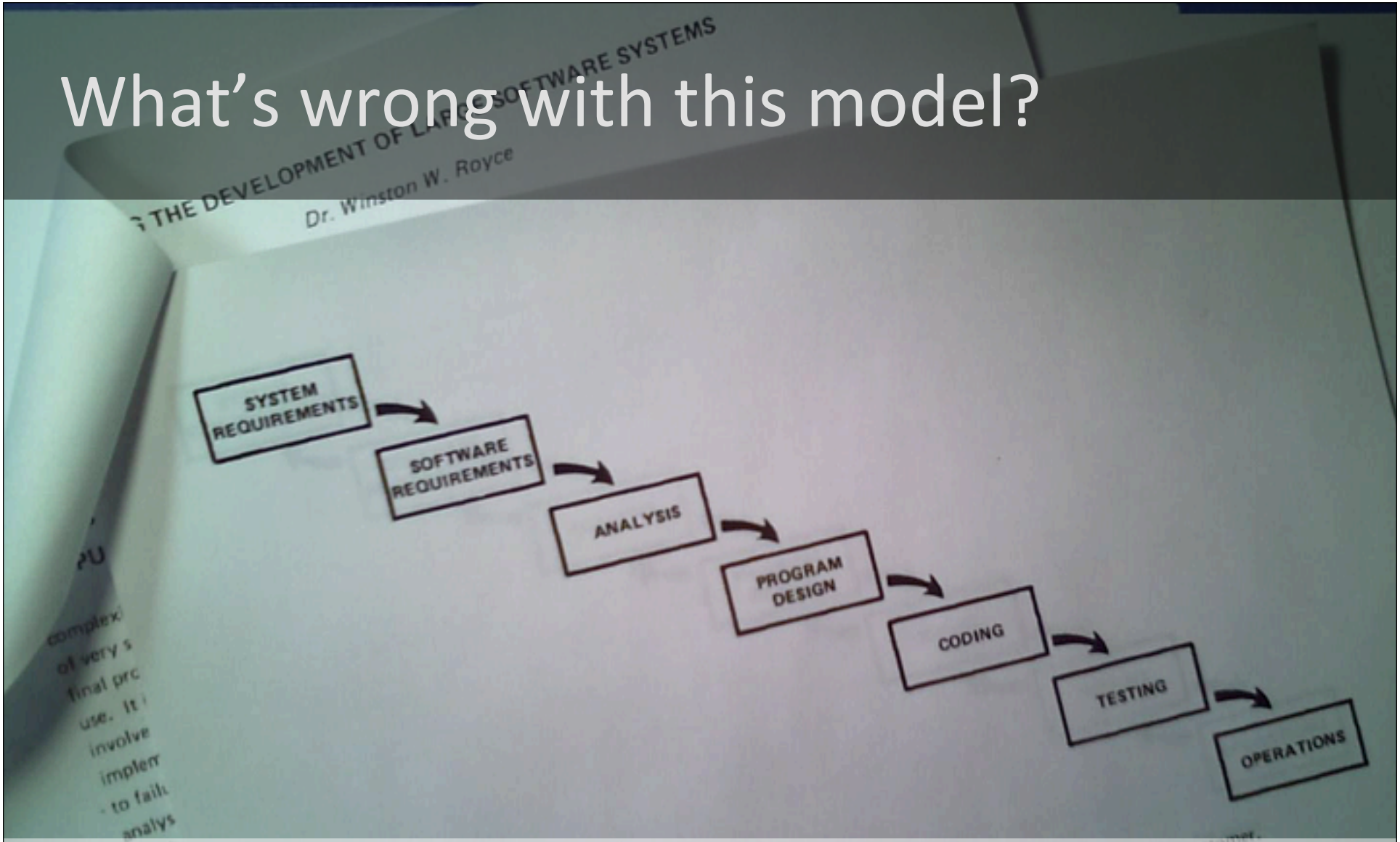


Process


THE DEVELOPMENT OF LARGE SOFTWARE
Dr. Winston W. Royce
SYSTEM REQUIREMENTS
SOFTWARE REQUIREMENTS
DESIGN
CODING
TESTING
OPERATIONS
Process

(cc) BY-NC-SA

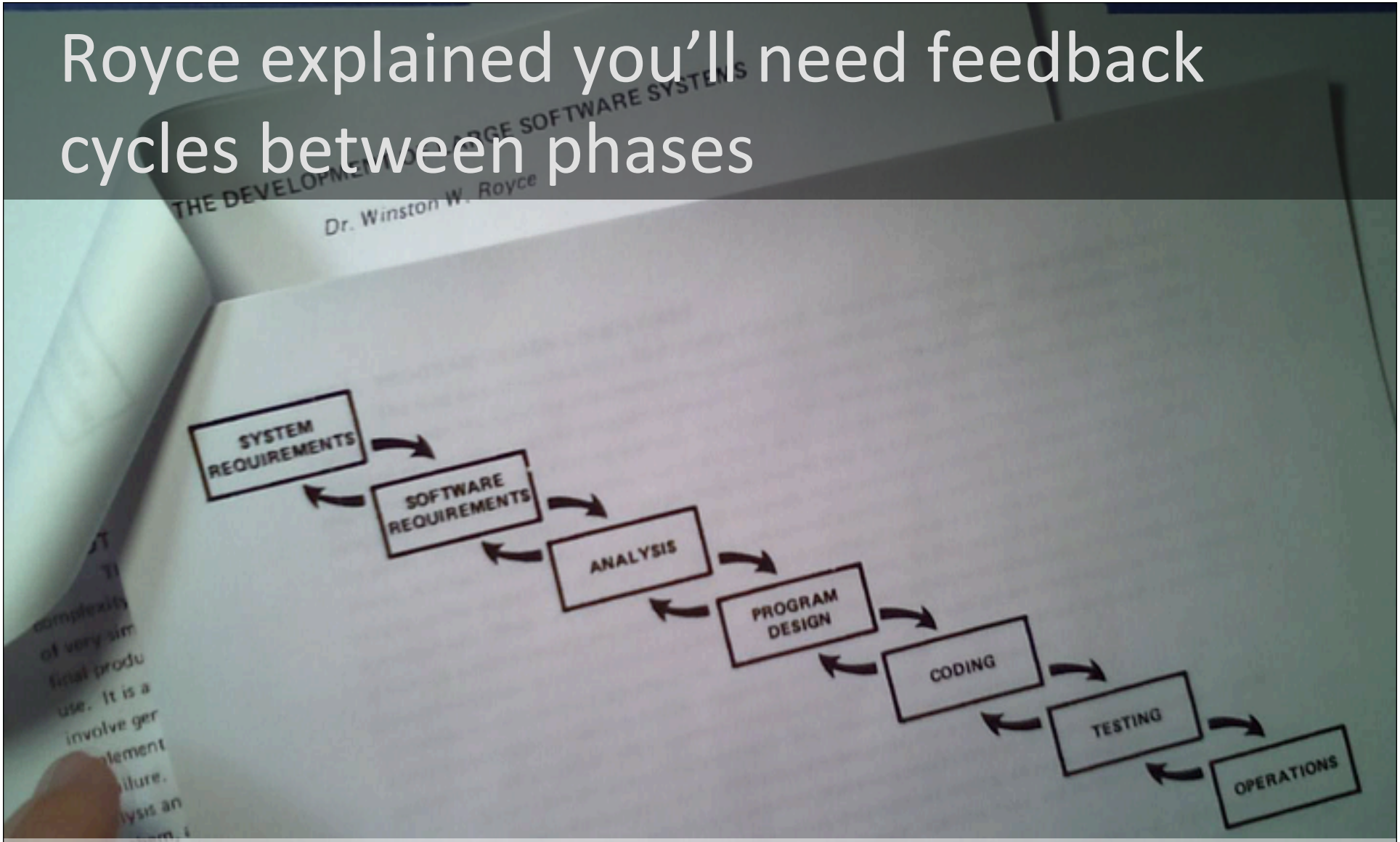
What's wrong with this model?



Winston Royce first published this model in 1970 as an interim figure on the way to his ideal approach

(cc) BY-NC-SA

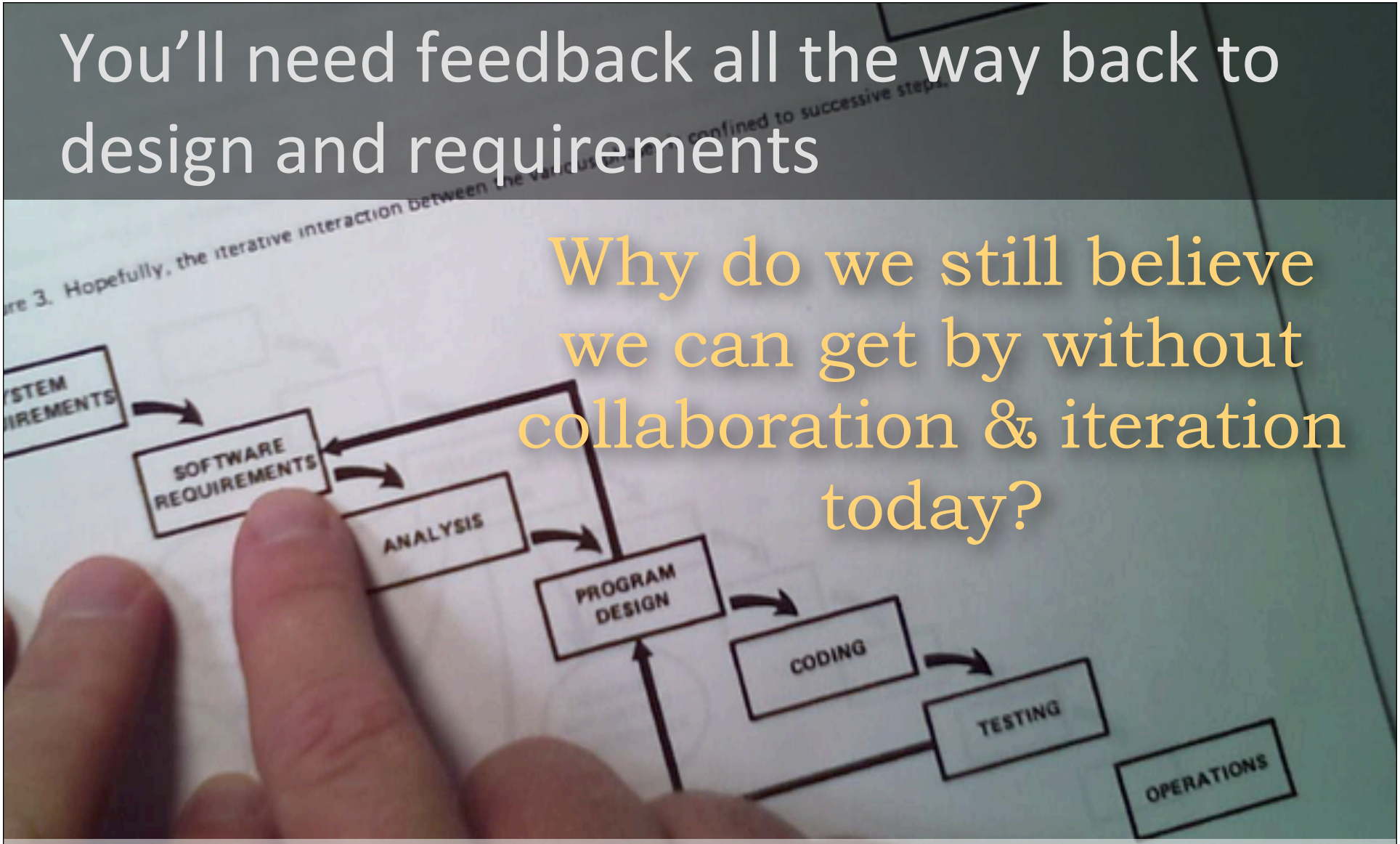
Royce explained you'll need feedback cycles between phases



Handing off documents and moving on doesn't work

(cc) BY-NC-SA

You'll need feedback all the way back to design and requirements

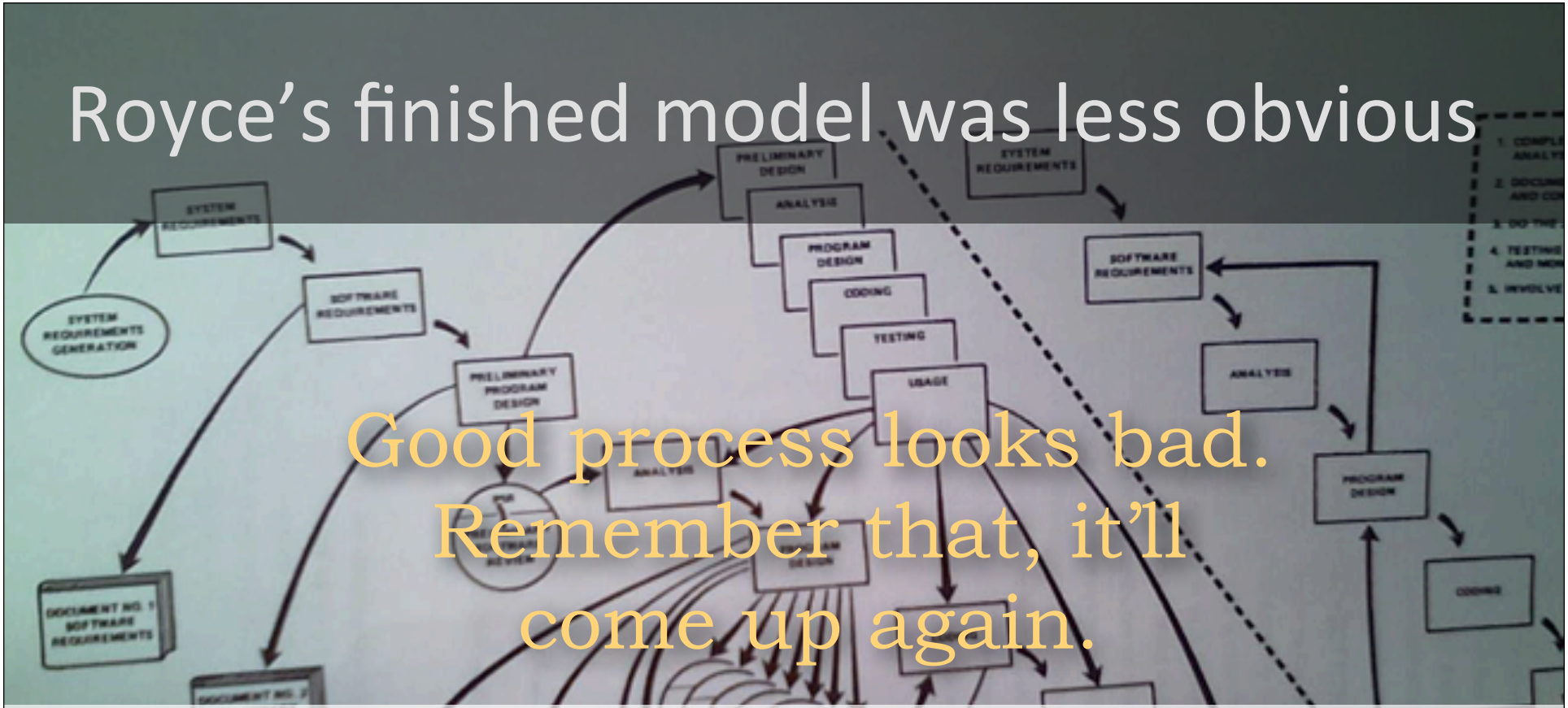


Why do we still believe we can get by without collaboration & iteration today?

It's only after testing that we understand if either design or requirements were good

(cc) BY-NC-SA

Royce's finished model was less obvious

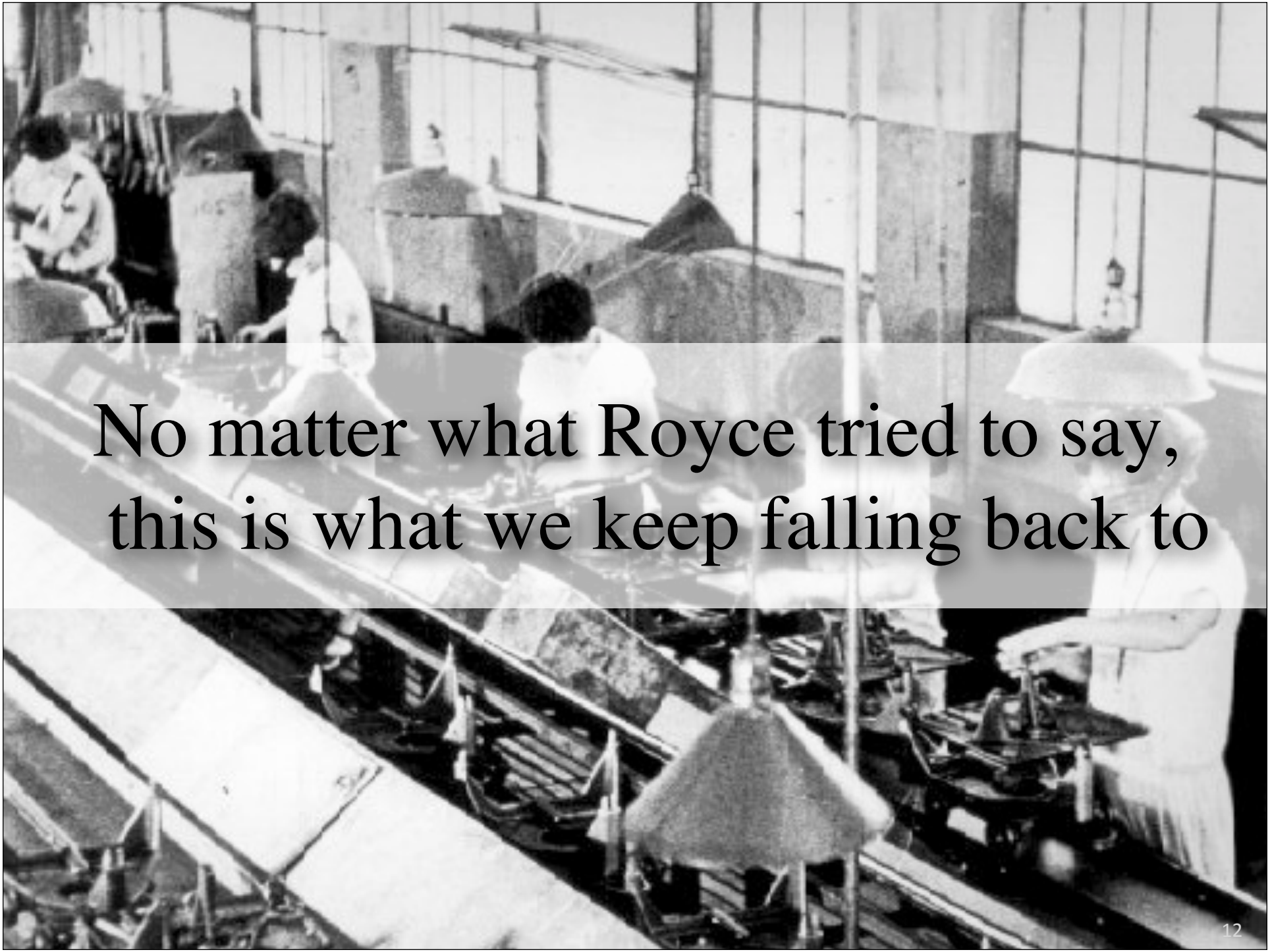


Good process looks bad. Remember that, it'll come up again.

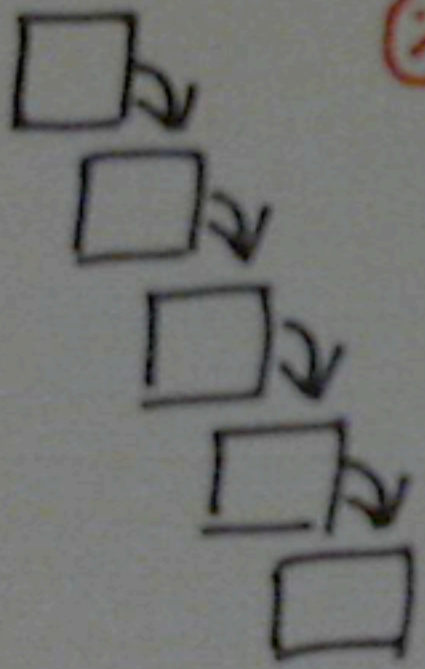
Royce's model had lots of collaboration, "involving the customer", recommended iteratively building the product - "do it at least twice."

"If the computer program in question is being developed for the first time, arrange matters so that the version finally delivered to the customer for operational deployment is actually the second version insofar as critical design/operations areas are concerned.

(cc) BY-NC-SA

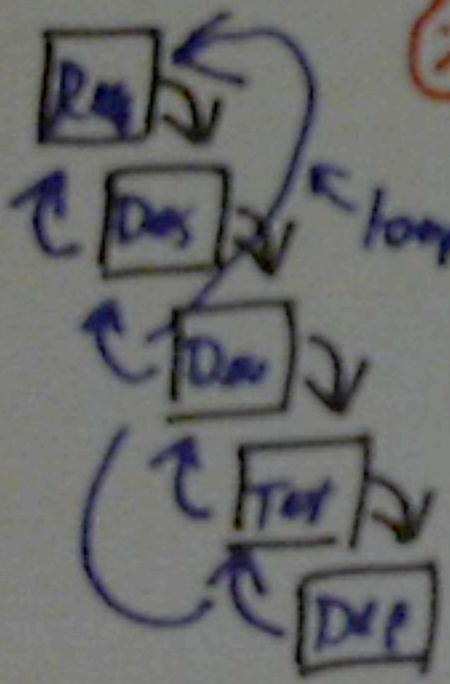

No matter what Royce tried to say,
this is what we keep falling back to

Royce's model



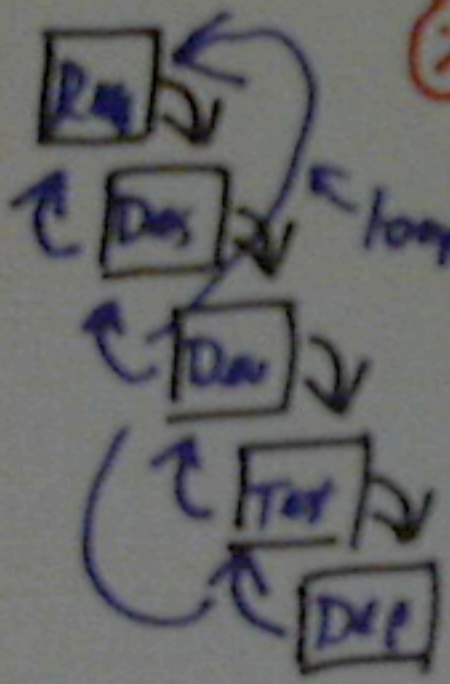
- takes too long
- requirements change before we get to the end
- delays in upstream activities squeeze downstream activities

Royce's model



- takes too long
- requirements change before we get to the end
- delays in upstream activities squeeze downstream activities

Royce's model



☹
- takes too long
- requirements change before we get to the end
- delays in upstream activities squeeze downstream activities

CYA

☺
- easy to understand
- easy to plan
- easy to track
- looks intuitive
- clear roles & responsibilities
- clear accountability

Agile Development will fix this, right?

(cc) BY-NC-SA

2011: “[In the UK], the Department for Work and Pensions (DWP) claims the IT behind its £2.2 billion flagship Universal Credit (UC) project is being developed using agile principles.”

“DWP gained agile experience using an unconventional approach developed by Accenture. It contracted lawyers to design contractual add-ons to accommodate the agile approach. Emergn, an agile systems supplier, developed an agile methodology that would apply to DWP's unique situation. Emergn refused to comment.”

What could possibly go wrong?

http://www.computerweekly.com/news/2240105764/Universal-Credit-deadline-forced-DWP-to-use-unproven-agile-development

(cc) BY-NC-SA

Agile kicked out

2013: “...in its final stages of testing for the pathfinder from April 2013, the programme is using the waterfall approach—a standard DWP testing methodology.” -- Mark Hoban, conservative member of parliament

A former principal agile consultant on Universal Credit, who asked not to be named, said:

"We were effectively on a waterfall project, because it was a waterfall contract,"

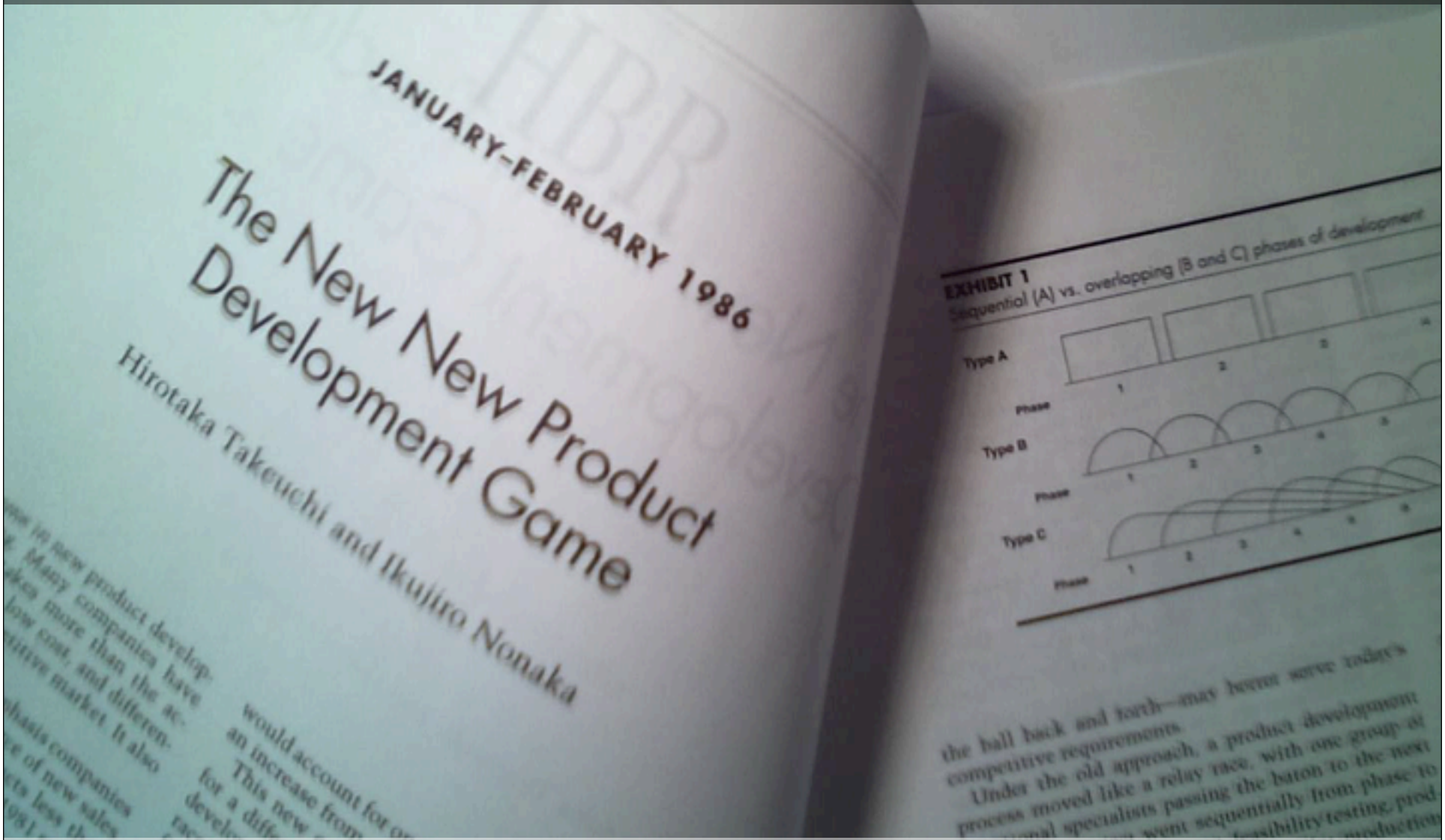
"You might call it agile, but if you don't set up the contract properly, it’s not. Call it what you like, but if it clucks and lays eggs, it’s a chicken."

(cc) BY-NC-SA

Good process looks messy

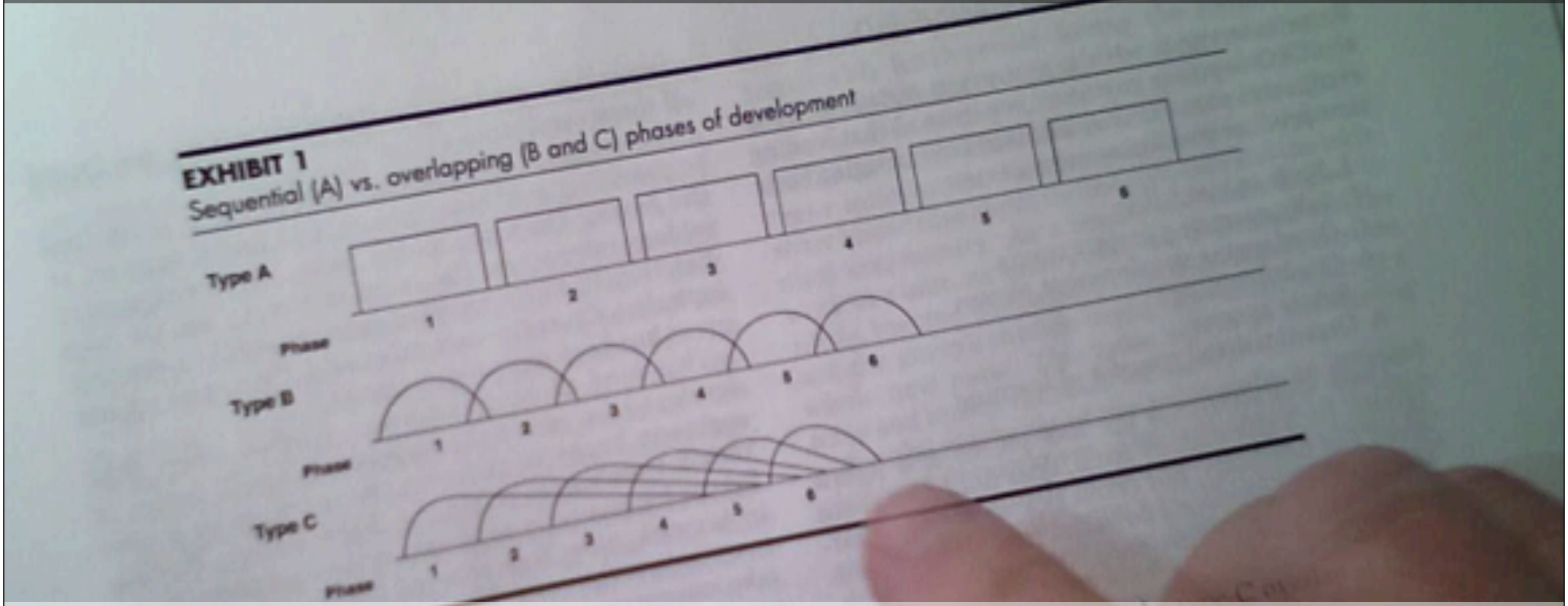
(cc) BY-NC-SA

Successful product companies use different approaches



(cc) BY-NC-SA

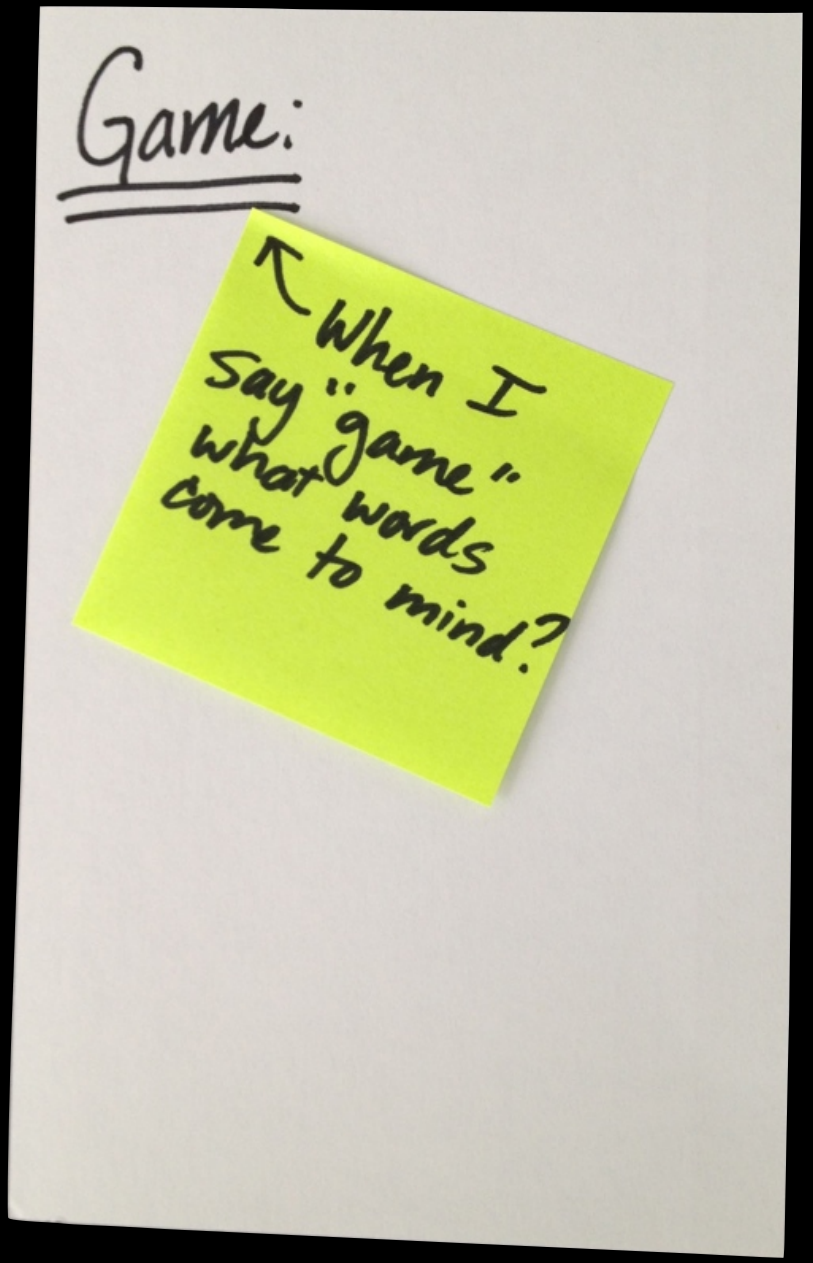
Takeuchi & Nonaka saw three process models



"Under the rugby approach, the product development process emerges from the constant interaction of a hand-picked, multidisciplinary team whose members work together from start to finish."

(cc) BY-NC-SA

What does “game” mean to you?



It’s all just a game

Games have

Simple Rules

Almost anyone can learn to play

The sophistication comes from **strategies and tactics** used by skilled players and coaches

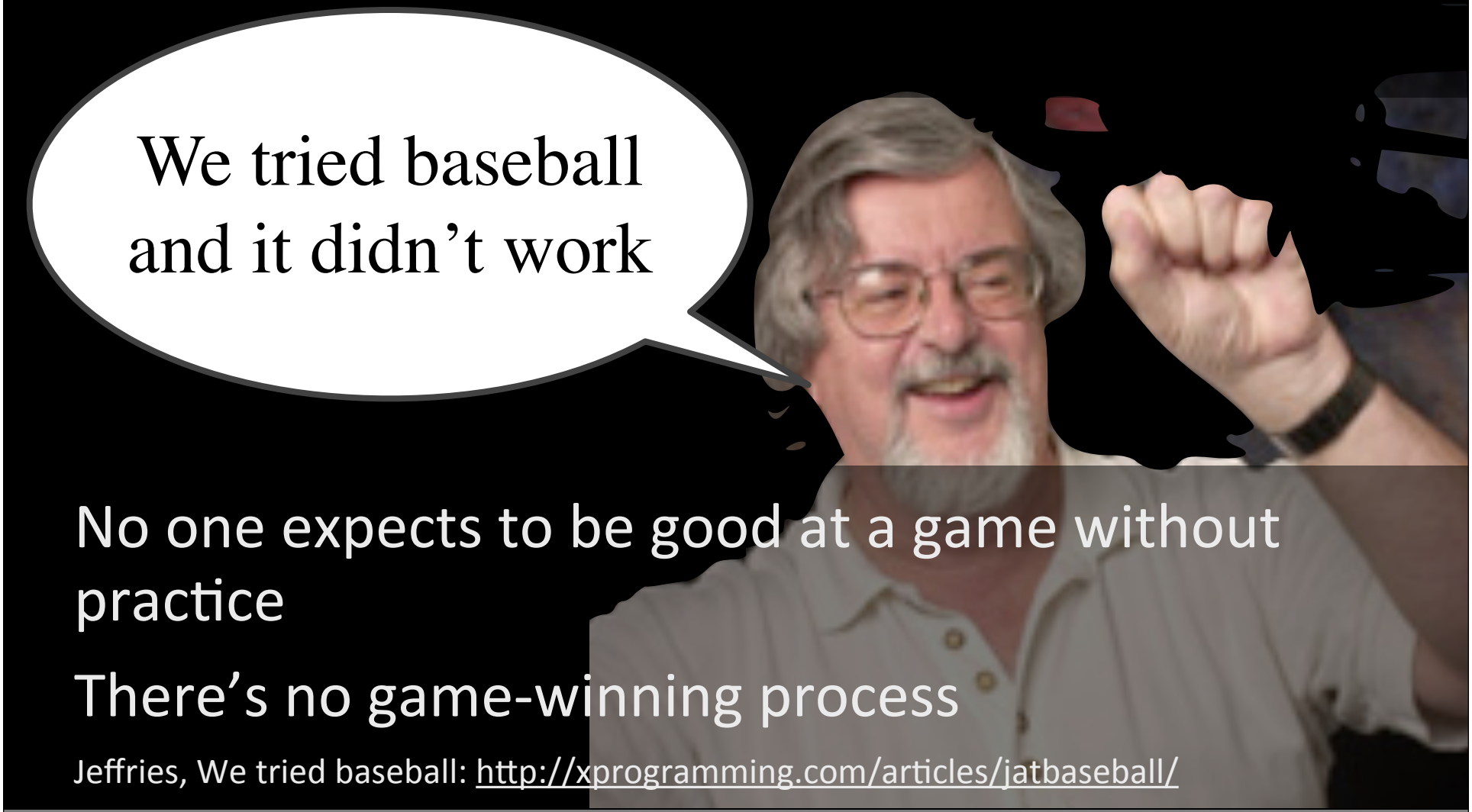
Games have

Positions not Roles

Players on a sports team build deep **specialization**

but **maintain general skills** to play many positions

Process ≠ skill

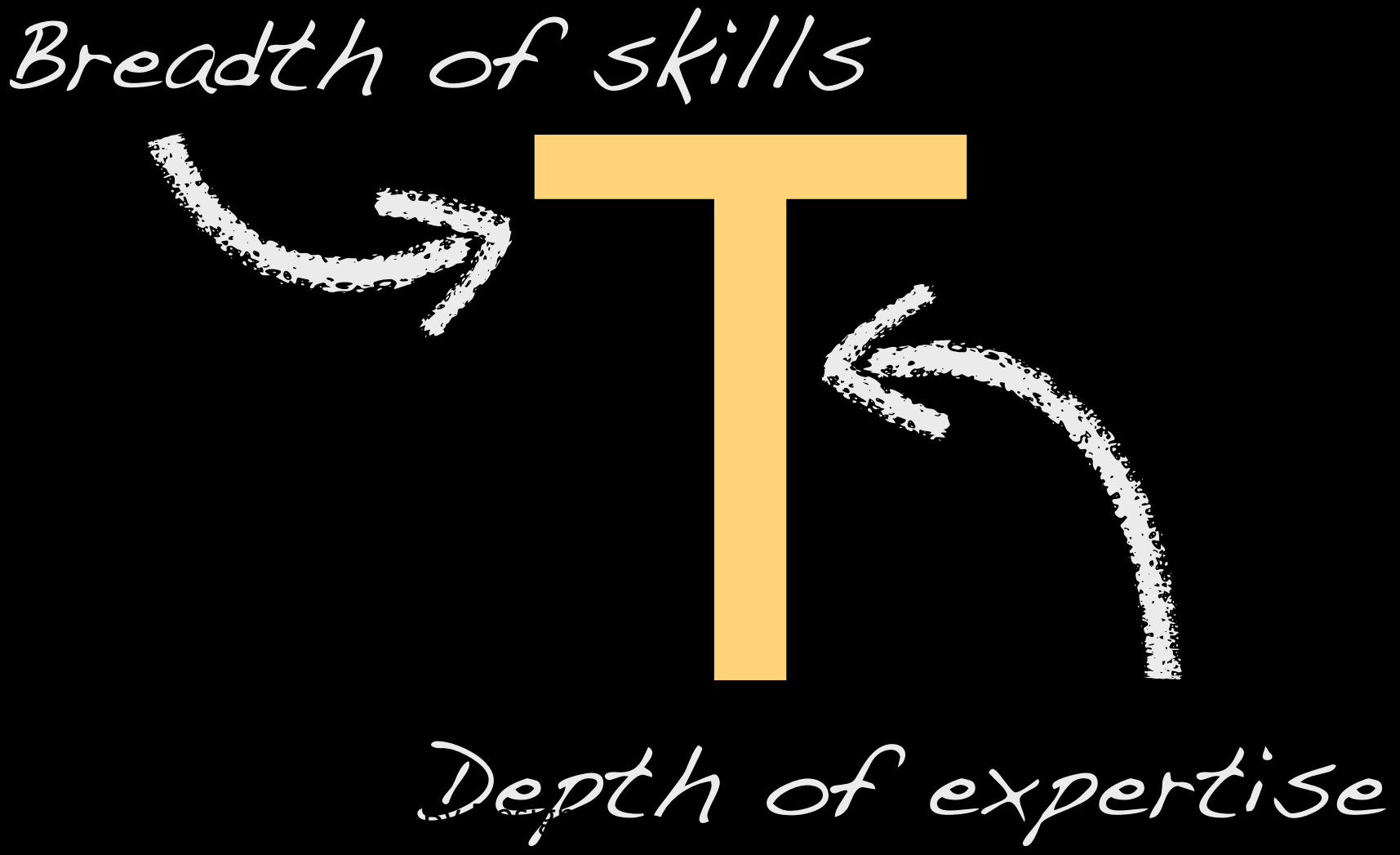


No one expects to be good at a game without practice

There's no game-winning process

Jeffries, We tried baseball: http://xprogramming.com/articles/jatbaseball/

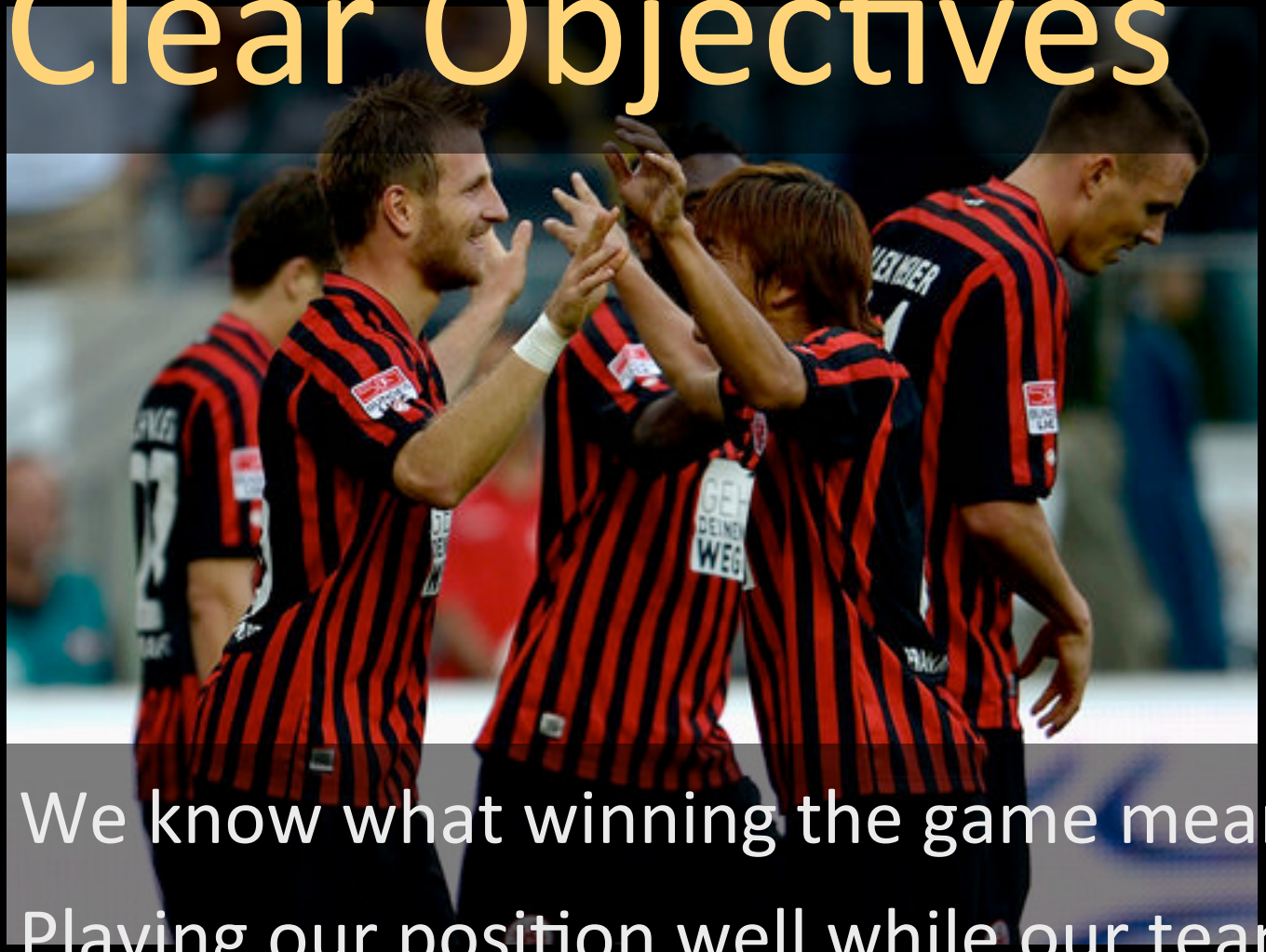
Build breadth and depth of skills



Build many skills

Wide breadth of skills along with varying depth in a variety of skills

Games have Clear Objectives



We know what winning the game means

Playing our position well while our team loses isn't considered success

output

outcome

we build this

we want this

© 2009-2011 Jeff Patton, all rights reserved, www.AgileProductDesign.com

Process ≠ Skill

knowing how doesn't make you good

Roles ≠ Positions

You might primarily do one thing, but you can't win by doing only one thing

Finishing On Time ≠ Winning

Keep score, don't just keep time

c

"Simple, clear purpose and principles give rise to complex and intelligent behavior.

Complex rules and regulations give rise to simple and stupid behavior."

-- Dee Hock

Keep rules to a minimum

Allow teams freedom to control their own strategies and tactics

We need enough rules to be playing the same game

Effective process is more like a team sport, and less like an assembly line

(cc) BY-NC-SA

Liquidnet plays like a single team

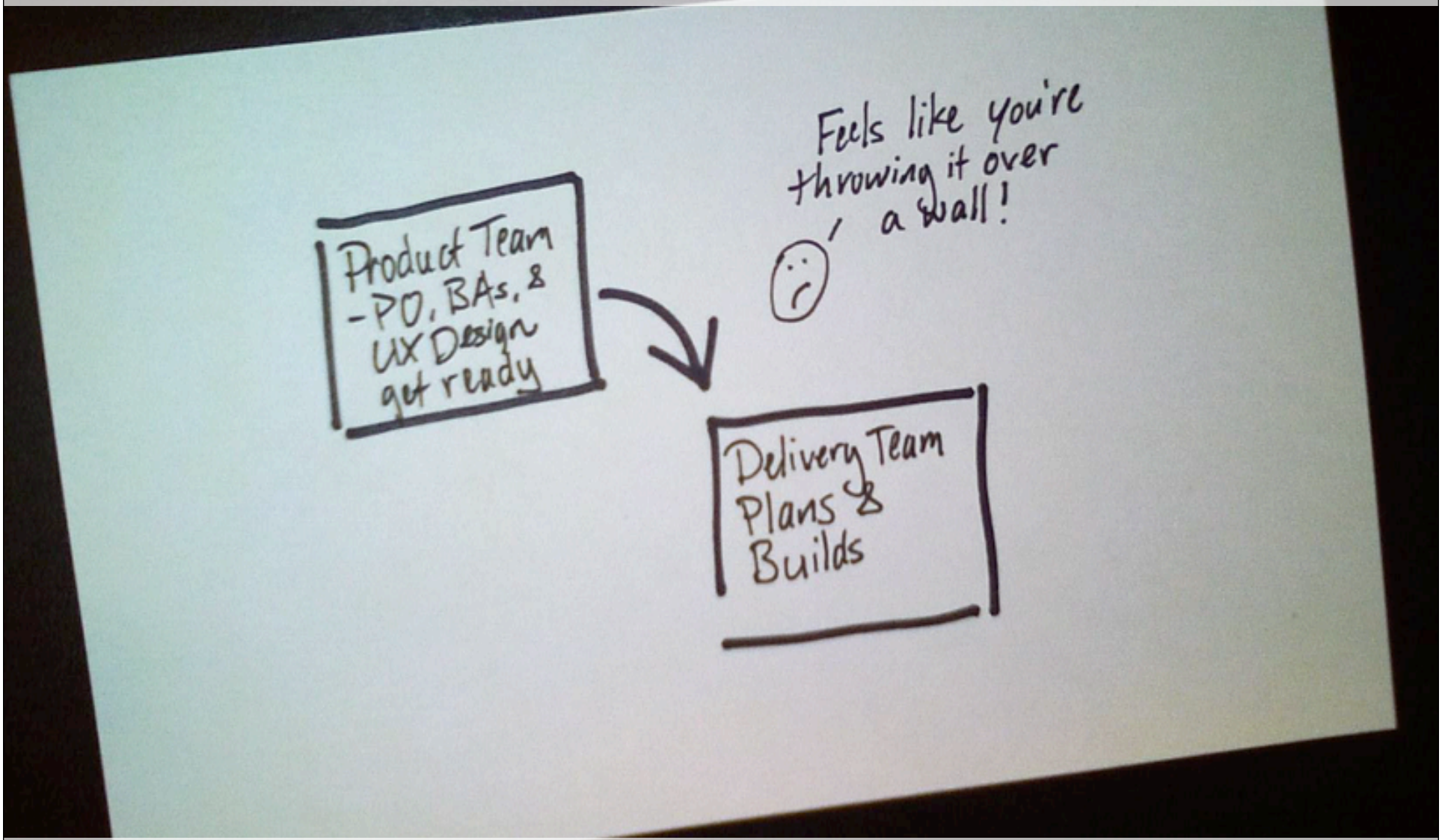
© 2009-2011 Jeff Patton, all rights reserved, www.AgileProductDesign.com

Liquidnet feature team at a daily standup

Liquidnet builds software for large institutional stock traders such as banks, hedge funds, pension funds, and mutual funds.

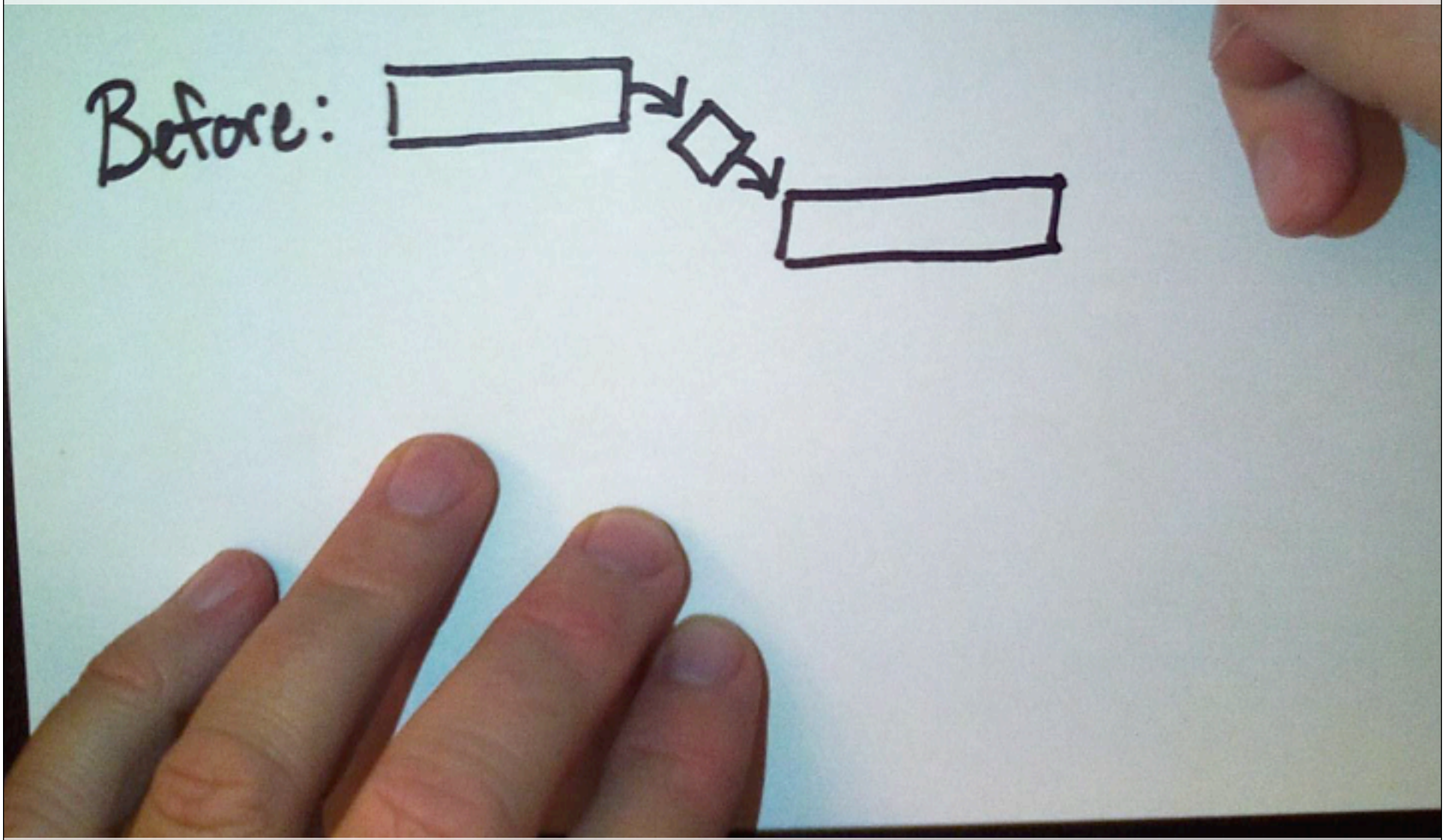
The next major revision of their flagship product use a single team with over 40 total team members and would take over a year to release.

The team used Scrum, but to them it felt like waterfall



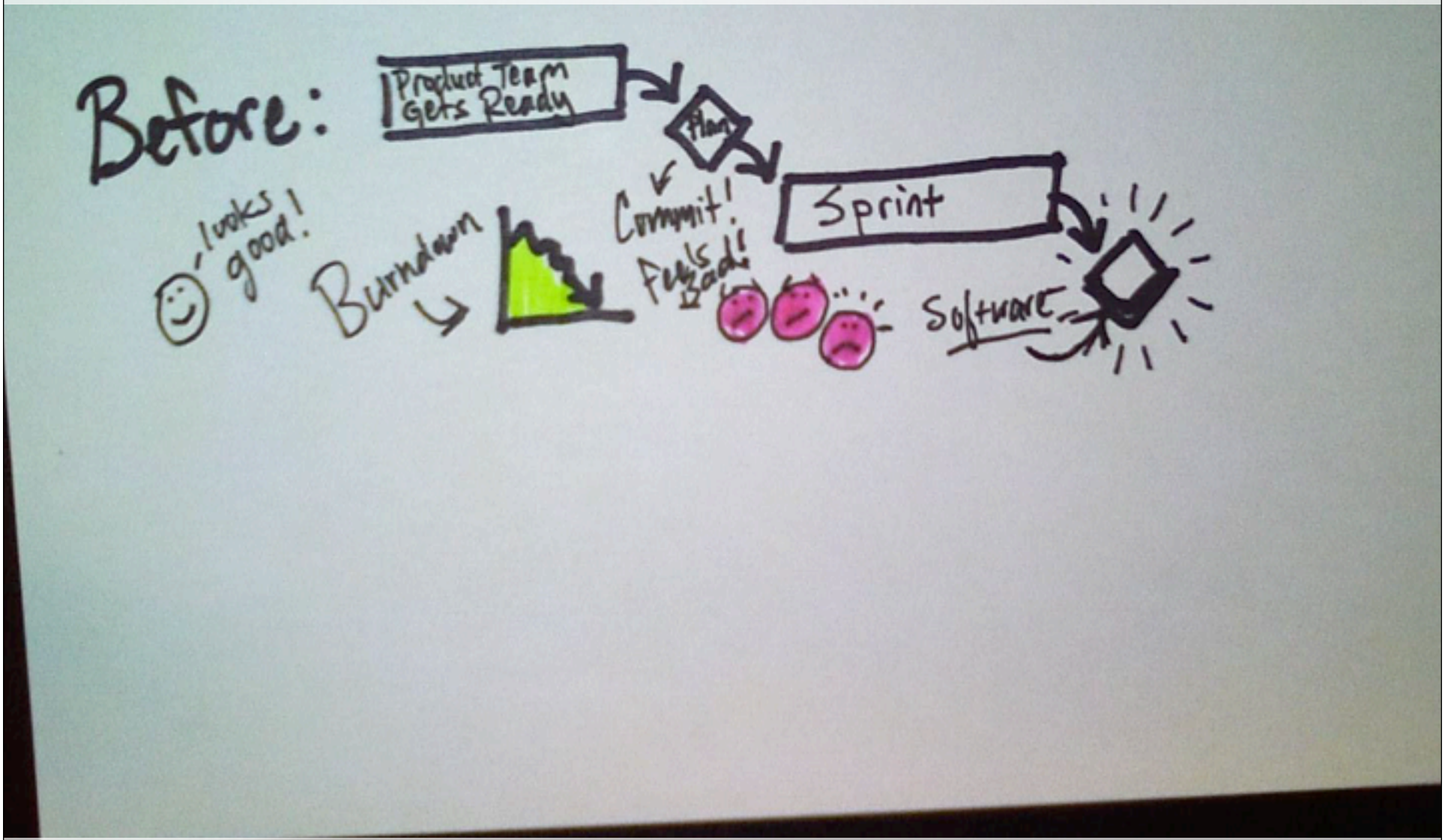
(cc) BY-NC-SA

They used a common staggered sprint model



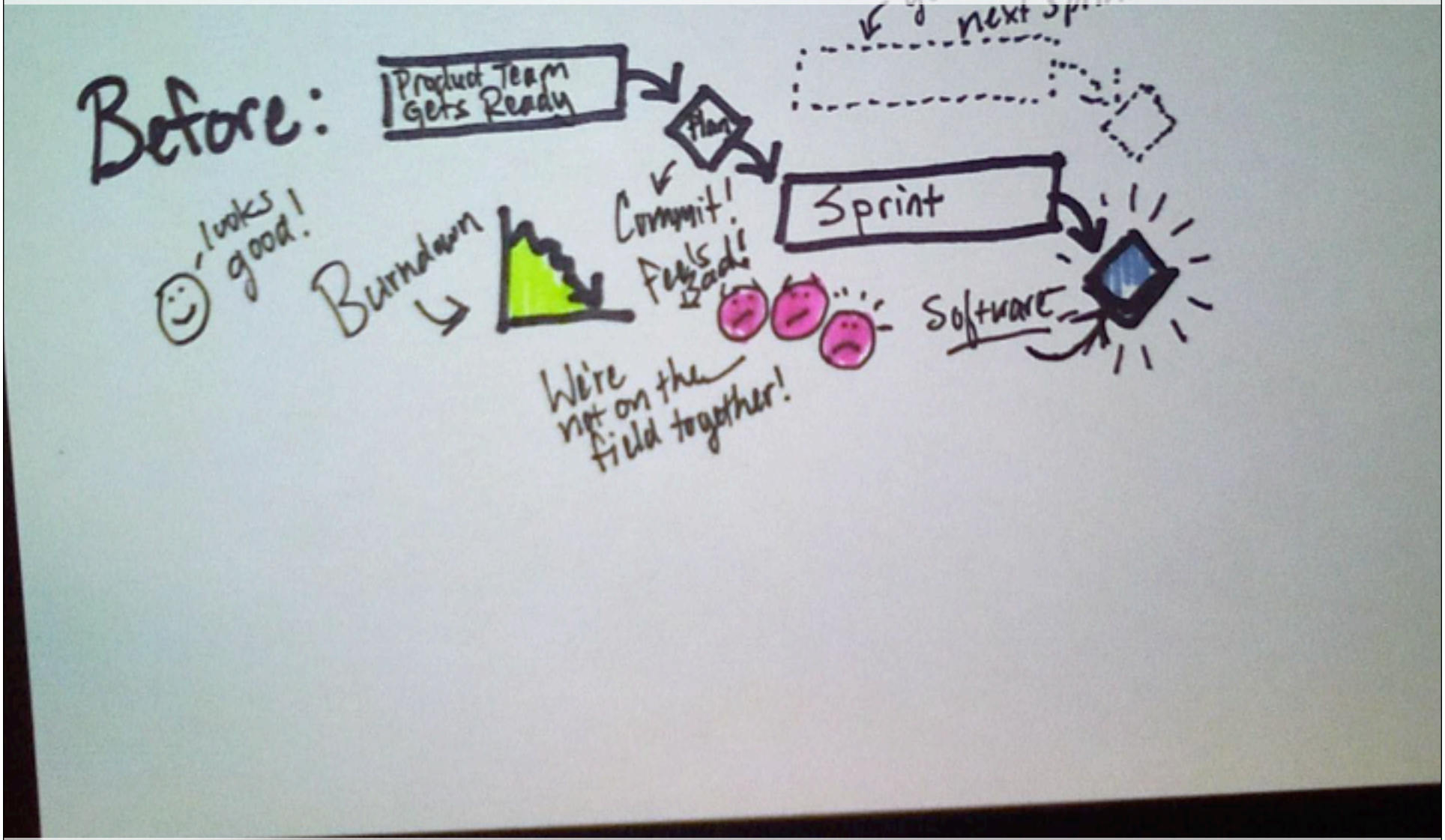
(cc) BY-NC-SA

A product owner, BAs and UX designers get stories ready for planning



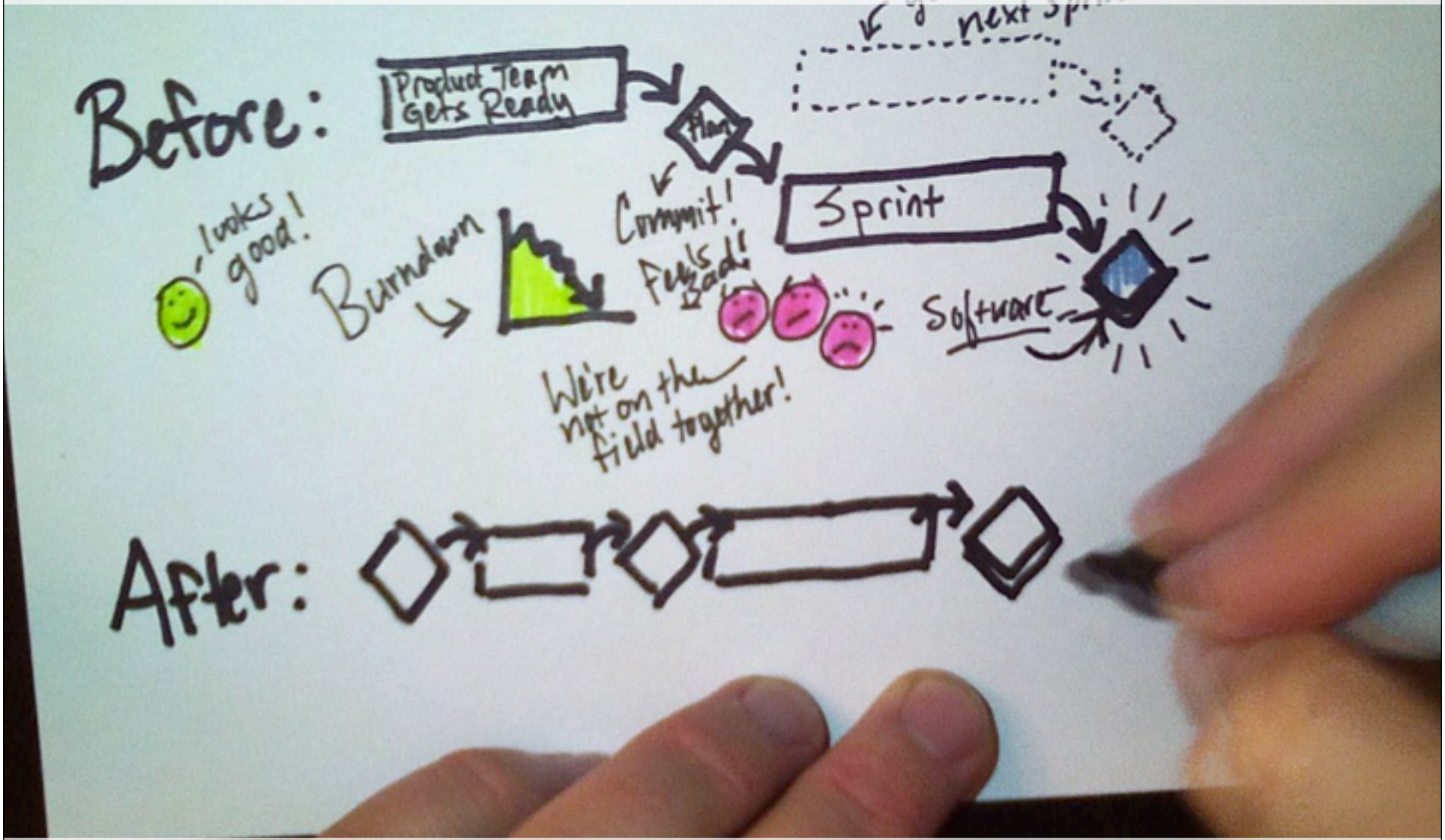
(cc) BY-NC-SA

While the team works, the product team get's ready for the next sprint



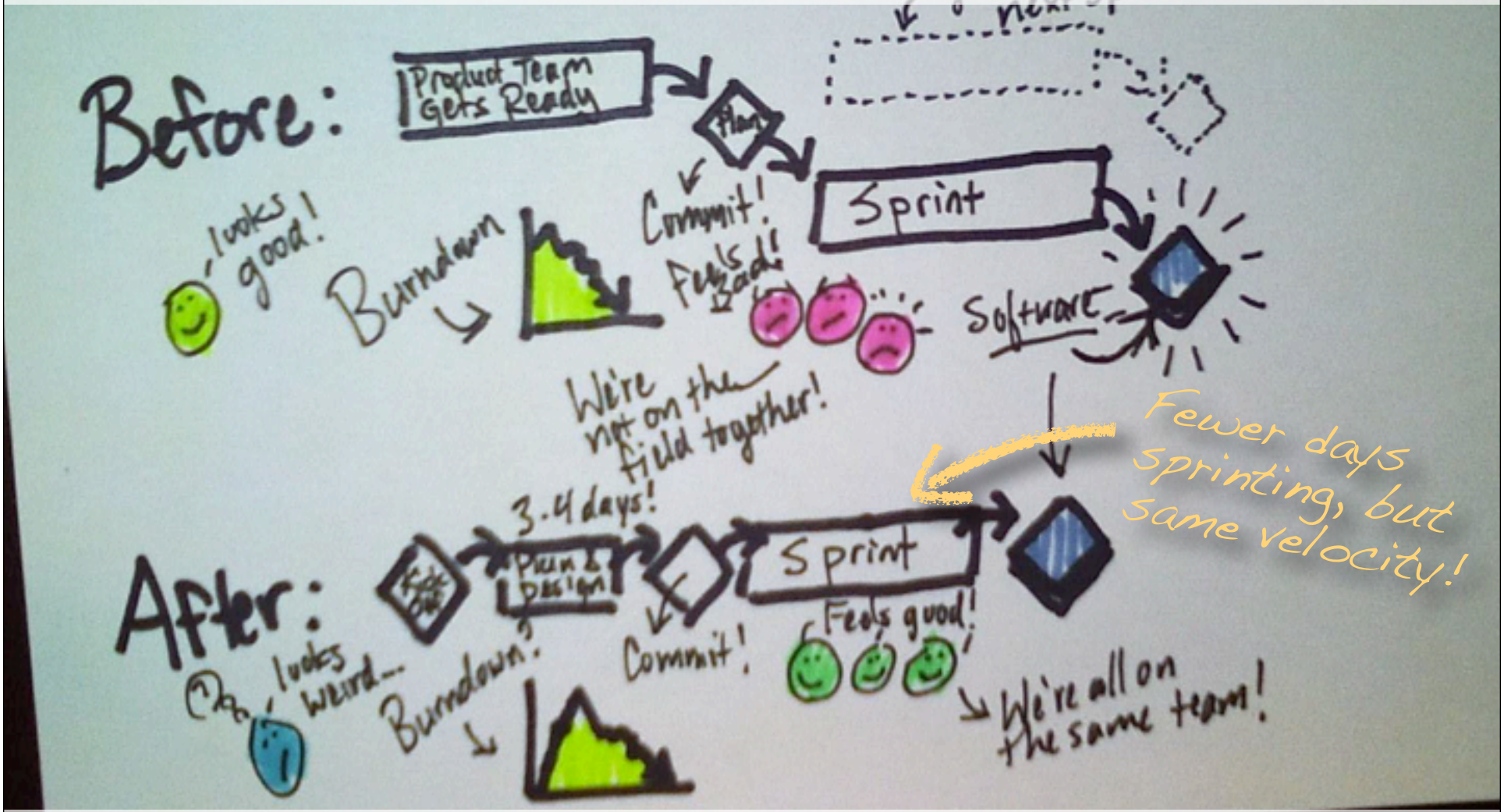
(cc) BY-NC-SA

The team alters their sprint to stay together



(cc) BY-NC-SA

The team gets ready together and sprints together



* 2009, Frank & Hartel, Feature Teams Collaboratively Building Products from READY to DONE

(cc) BY-NC-SA

Scrum is an agile strategy, not the rules of the game

(cc) BY-NC-SA

But

(cc) BY-NC-SA

Even if you are on time, you're probably wrong anyway

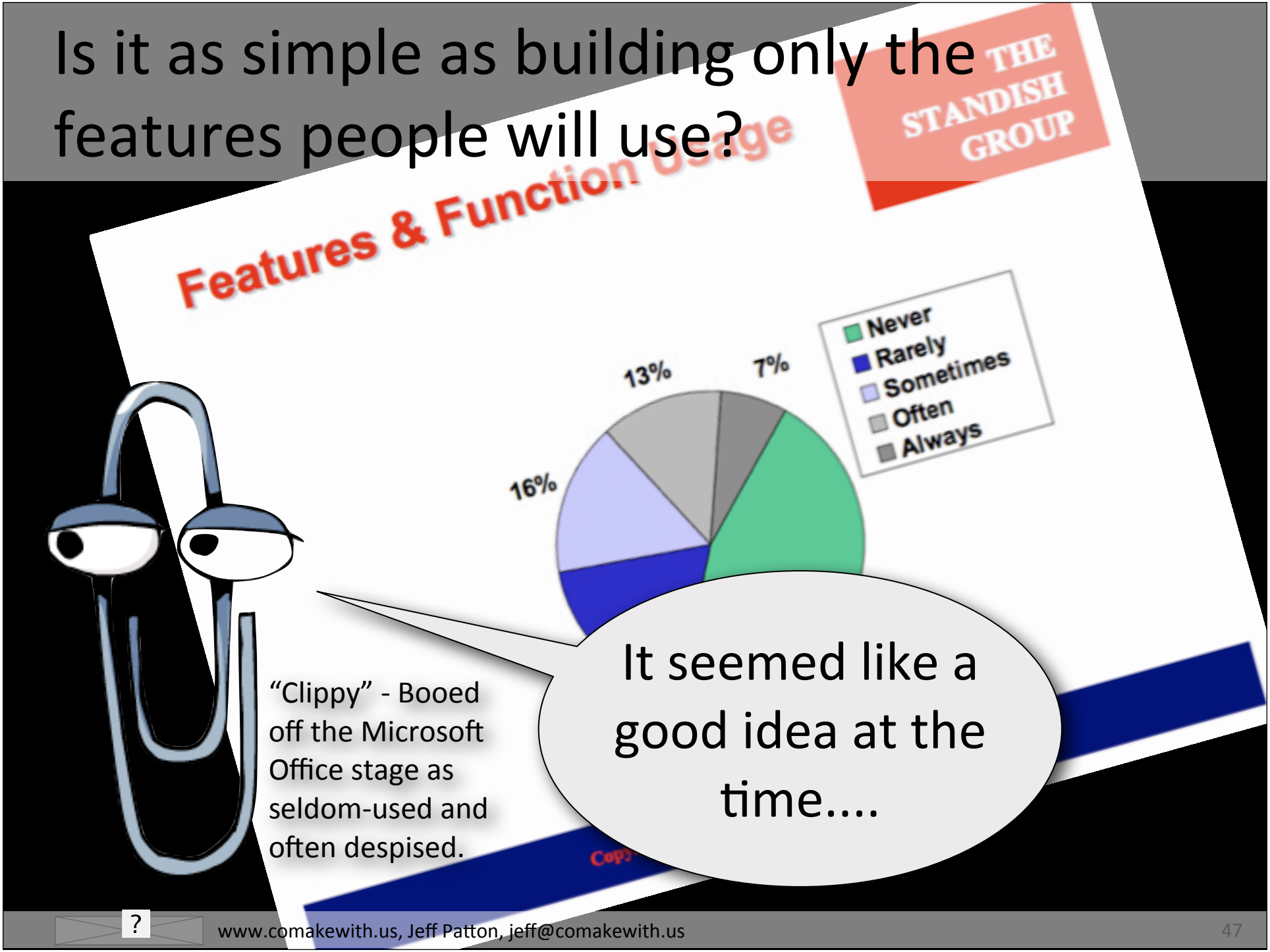
(cc) BY-NC-SA

Your guesses about the future are probably wrong

People like Marty say this stuff is hard

(Marty Cagan, author of **Inspired**, How to Create Products Customers Love)

Is it as simple as building only the features people will use?



It's only after delivery that we really understand value

opportunity: integrated mus management a portable musi player

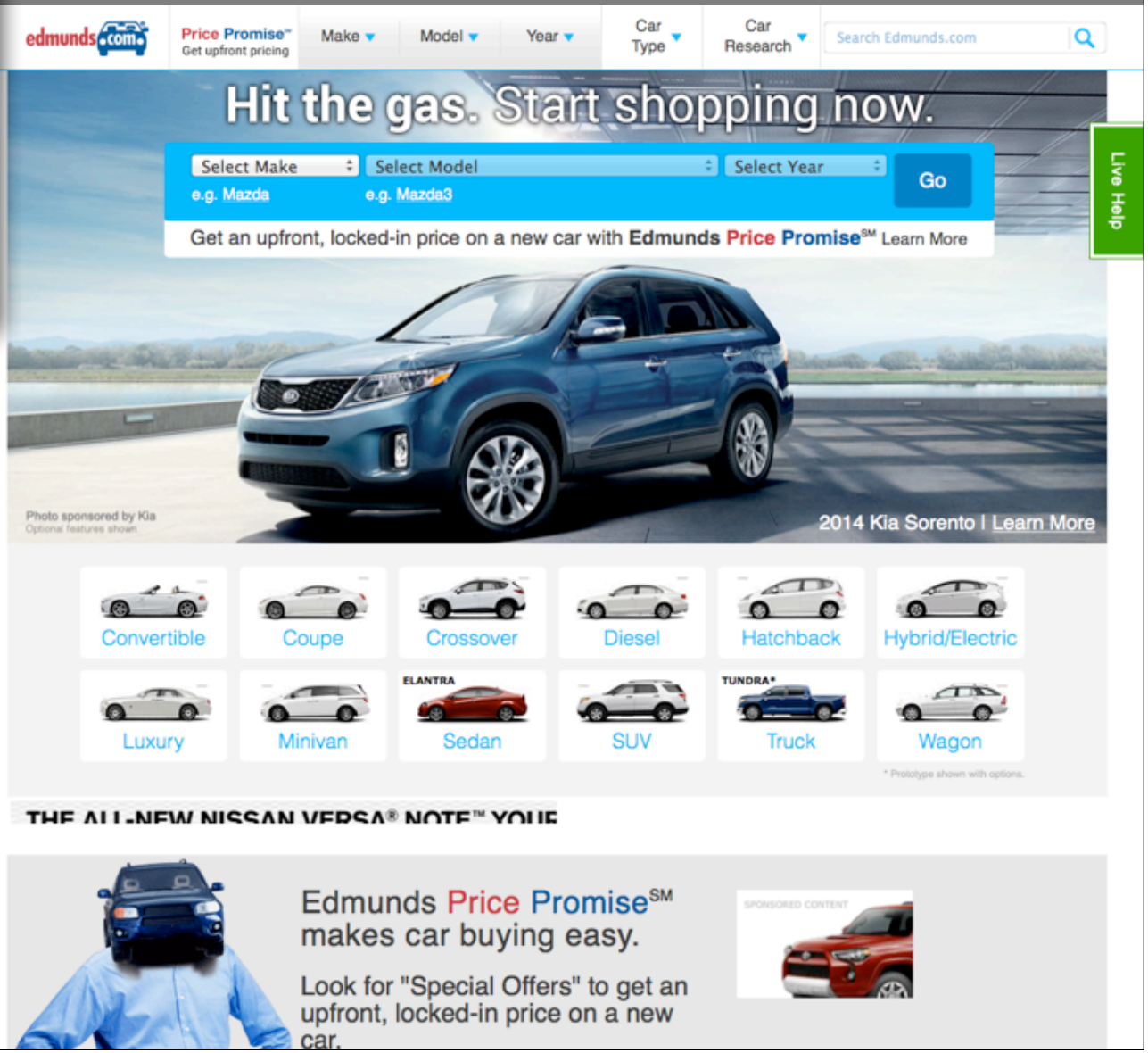
Adding design thinking to fix agile

(cc) BY-NC-SA

Edmunds.com has mastered Agile, but it doesn't matter

Eugene Park is the director of product management at Edmunds.com

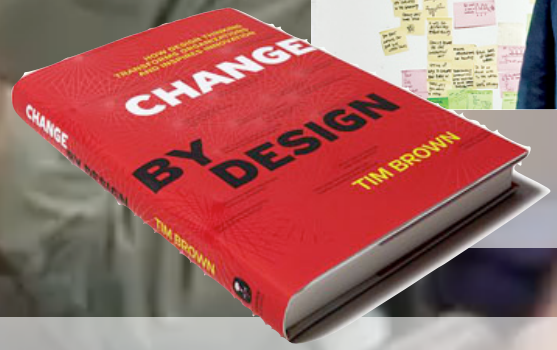
Edmunds was founded in 1966 as a publisher of printed booklets consolidating automotive specifications to help car shoppers make buying decisions. Today they're one of the most trusted sources on the web for auto buyers and enthusiasts.



We'd like to adopt design thinking

Design thinking focuses whole teams on solving customer and user problems

IDEO Pioneered Design Thinking, schools like Standford's d.school and the HPI School of Design Thinking teach it

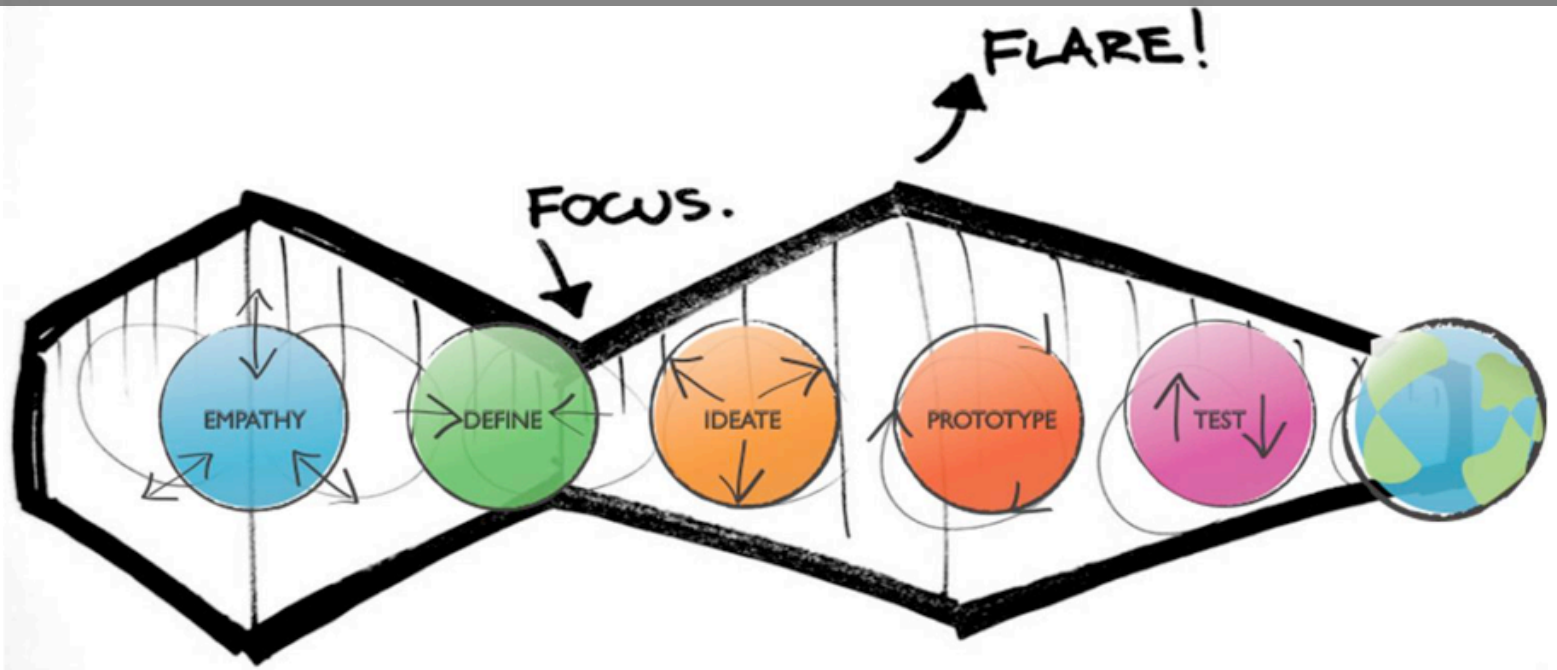


IDEO's Tim Brown, author of Change by Design

IDEO's "Shopping Cart Video": https://www.youtube.com/watch?v=M66ZU2PCIcM

(cc) BY-NC-SA

Design & Design Thinking



Design follows a basic flow from research to solution design

[It can sometimes be seen as an "ivory tower" activity for select individuals]

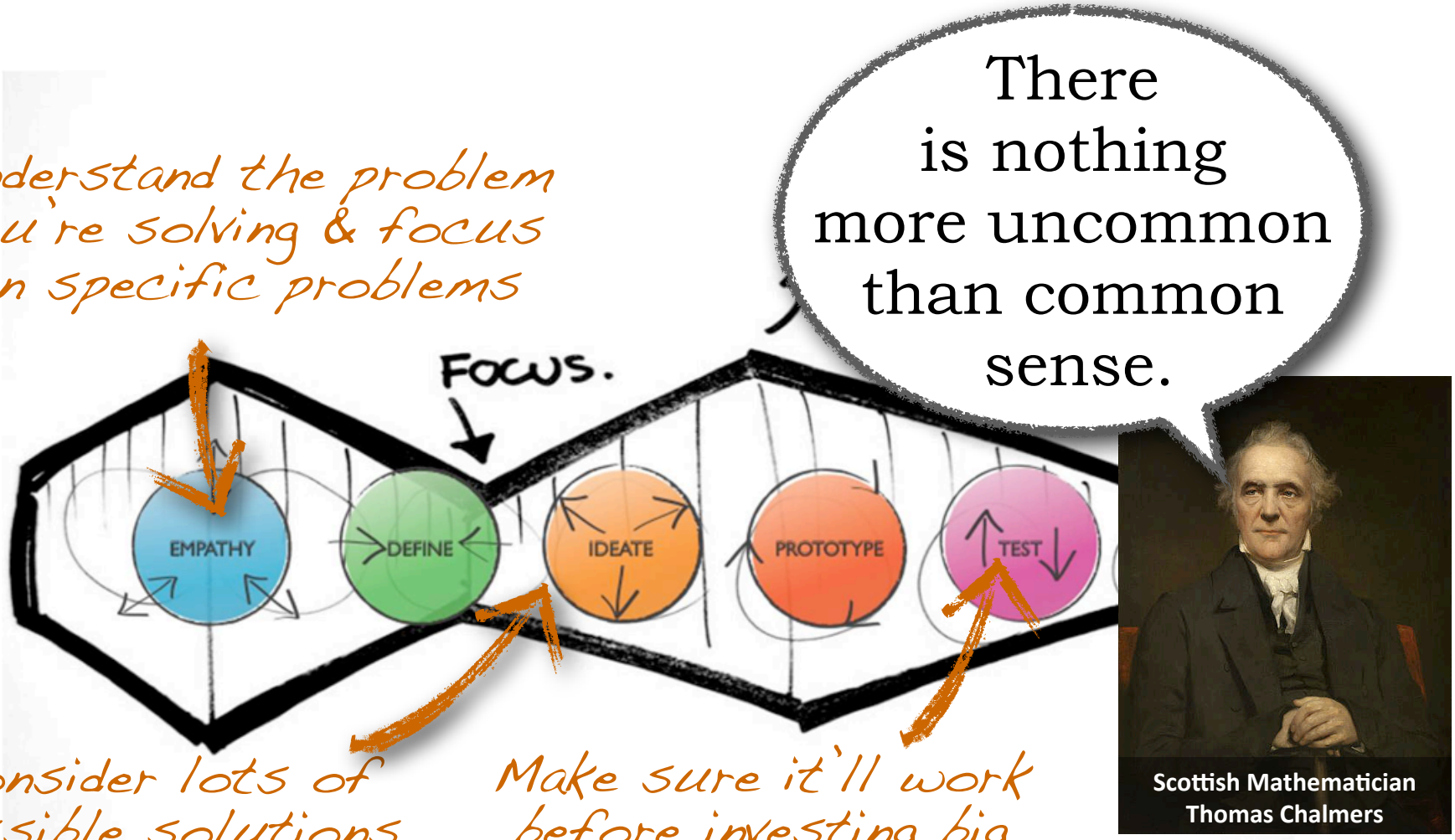
Design thinking leverages diverse teams, lots of collaboration, and user participation

[Design thinking asks everyone to participate, but still relies on strong leadership]

(cc) BY-NC-SA

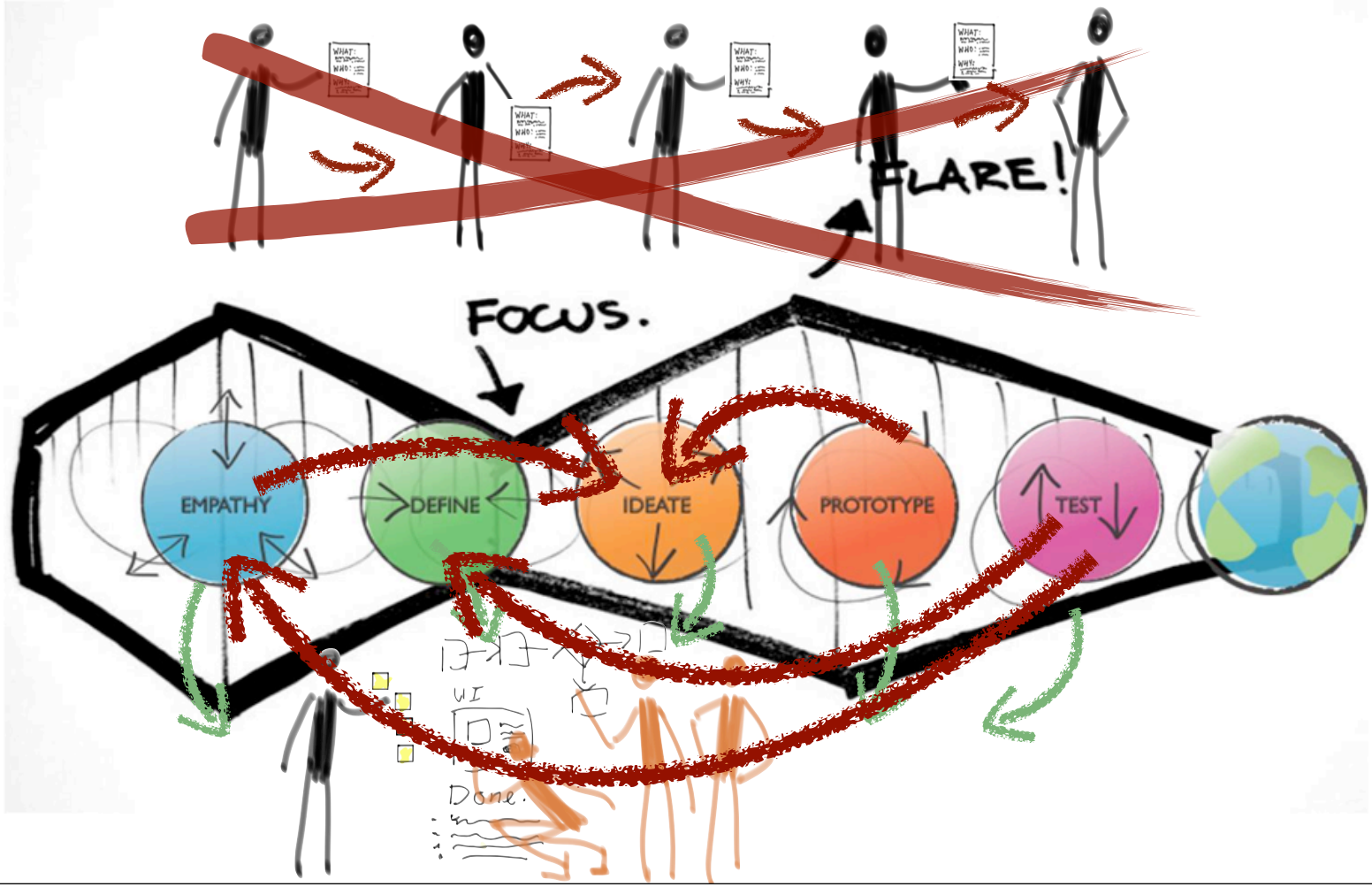
Design thinking seems like common sense, right?

There is nothing more uncommon than common sense.



Scottish Mathematician Thomas Chalmers

Different ways of thinking - NOT process roles, hand-offs, sequence and phases



Designers act as experts and guides

Leah Buley
www.adaptivepath.com/aboutus/leah.php

"Design isn't a product that designers produce, design is a process that designers facilitate."
-- Leah Buley

(cc) BY-NC-SA

(Almost) everyone at Edmunds spends time with users

(cc) BY-NC-SA

Teams work together to make sense of what they've learned

(cc) BY-NC-SA

They ideate by sketching independently

(cc) BY-NC-SA

Everyone shares their results

(cc) BY-NC-SA

Everyone shares their results

(cc) BY-NC-SA

They start to imagine solutions using paper, scissors, and tape

They start to imagine solutions using paper, scissors, and tape

Edmunds.com shares the product's story for all teams in an internal "trade show"

(cc) BY-NC-SA

Edmunds.com shares the product's story for all teams in an internal "trade show"

(cc) BY-NC-SA

Edmunds.com shares the product's story for all teams in an internal "trade show"

(cc) BY-NC-SA

Edmunds.com shares the product's story for all teams in an internal "trade show"

(cc) BY-NC-SA

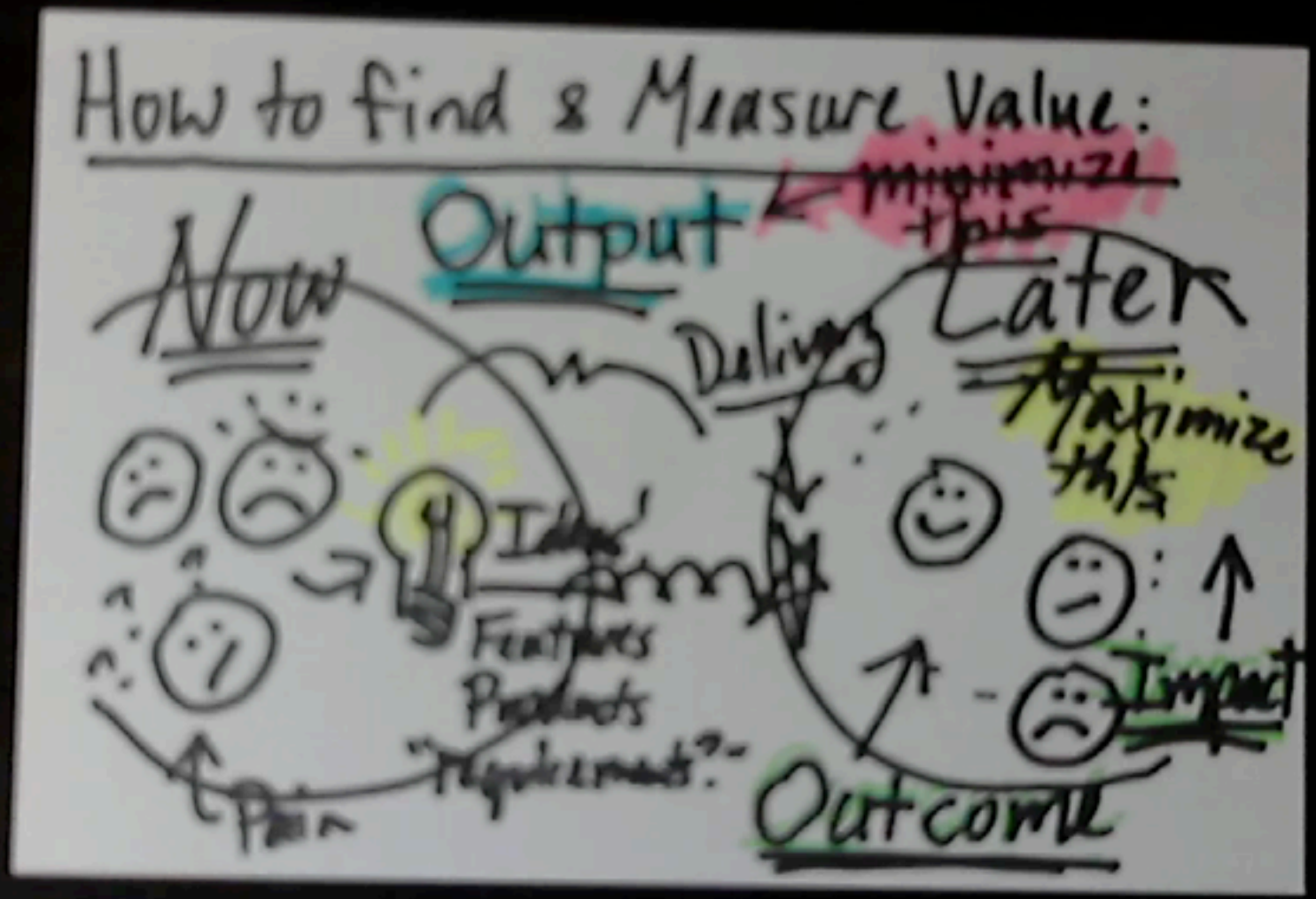
We've found simple ideas that now generate millions in revenue every year

Edmunds teams stopped worrying about velocity and started focusing on success

The secret to effective software development

It’s not about software

It’s about changing the world


How to find & Measure Value:
Output
minimize this
Now
Later
Delivery
Maximize this
Ideas
Features
Products
"Requirements?"
Pain
Impact
Outcome

We've gotten a lot
better at measuring outcomes

We're
still wrong most of the
time, but now we pull out
software that's not getting us
the outcomes we want.

Adding Lean Startup to fix Agile

Validated learning over working software (or comprehensive documentation)

In May of 2010, Kent Beck gave a keynote speech at the first Lean Startup conferences where he revised the agile manifesto from his perspective

Working software isn't enough to ensure product success

(cc) BY-NC-SA

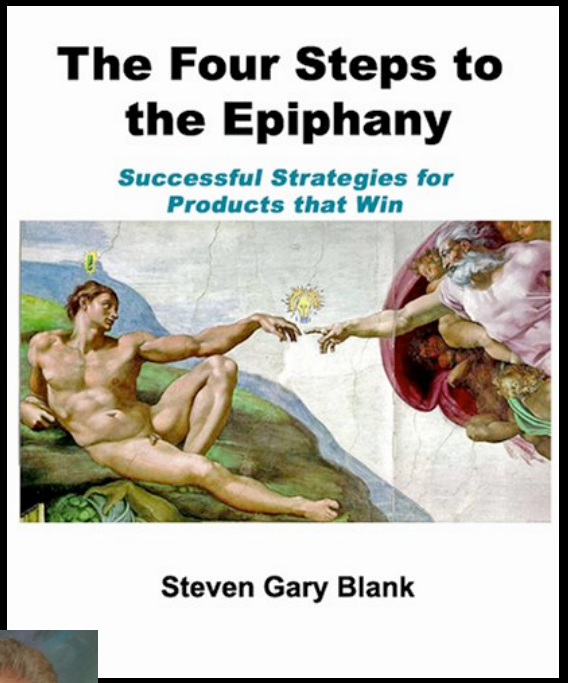
Does anyone want your product?

Customer Discovery

- Find customers who have a problem your product can solve

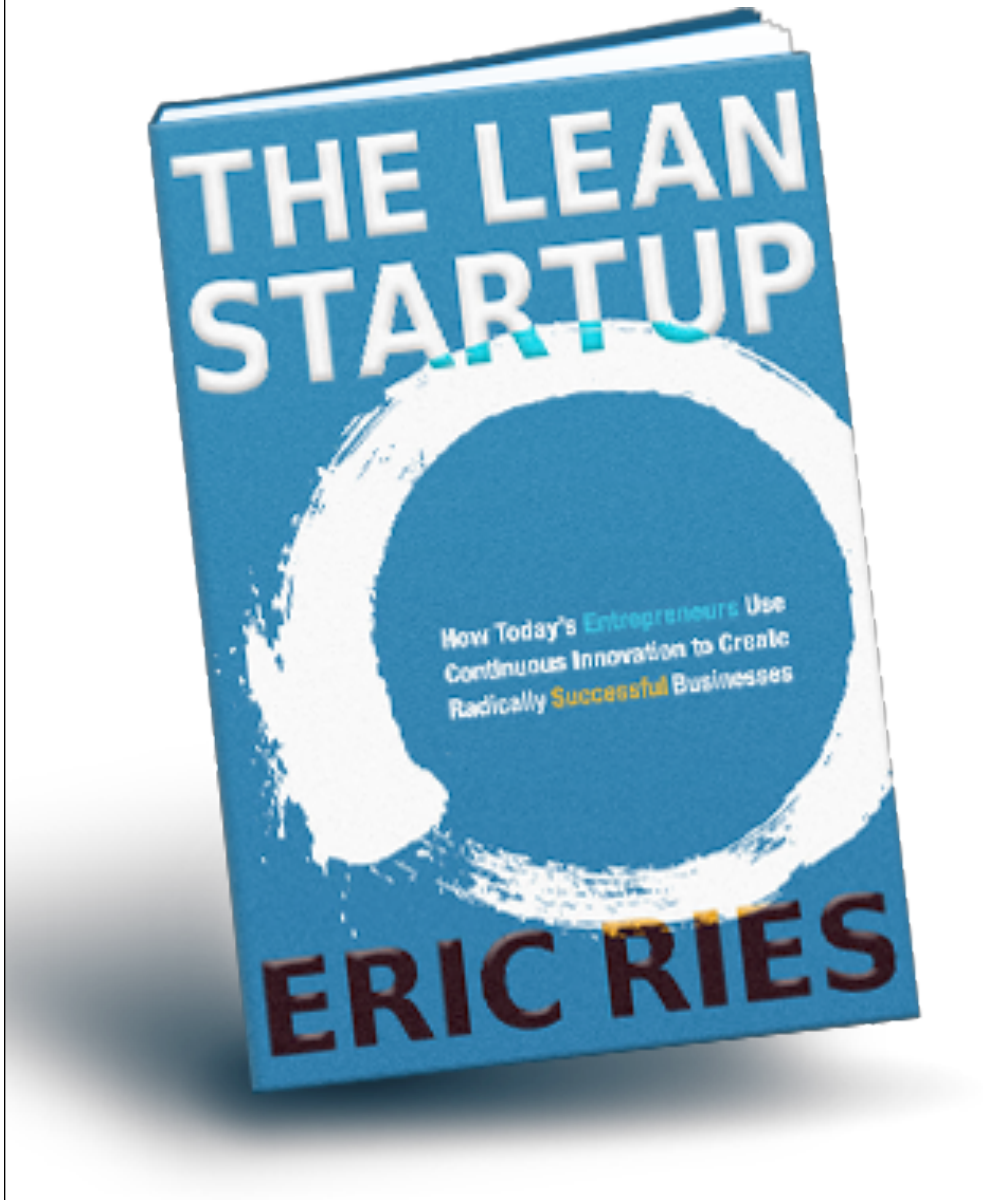
Customer Development

- Iteratively test solutions with target customers until you have a solution they'll buy and recommend to others



Steve Blank, author of The Four Steps to Epiphany

(cc) BY-NC-SA

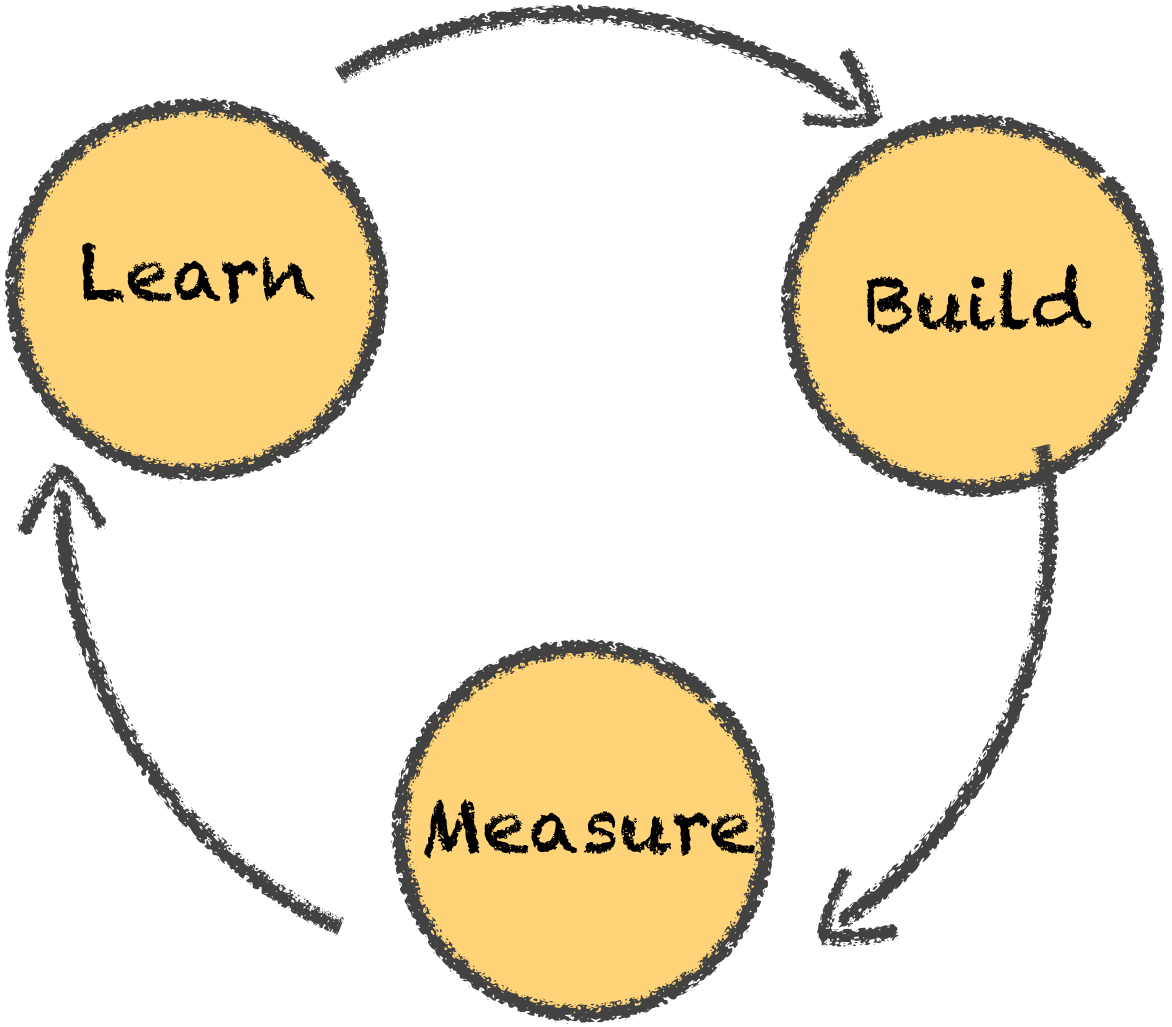


Eric Ries, author of The Lean Startup

Important concepts

Dumb title

The Lean Startup Validated Learning Loop



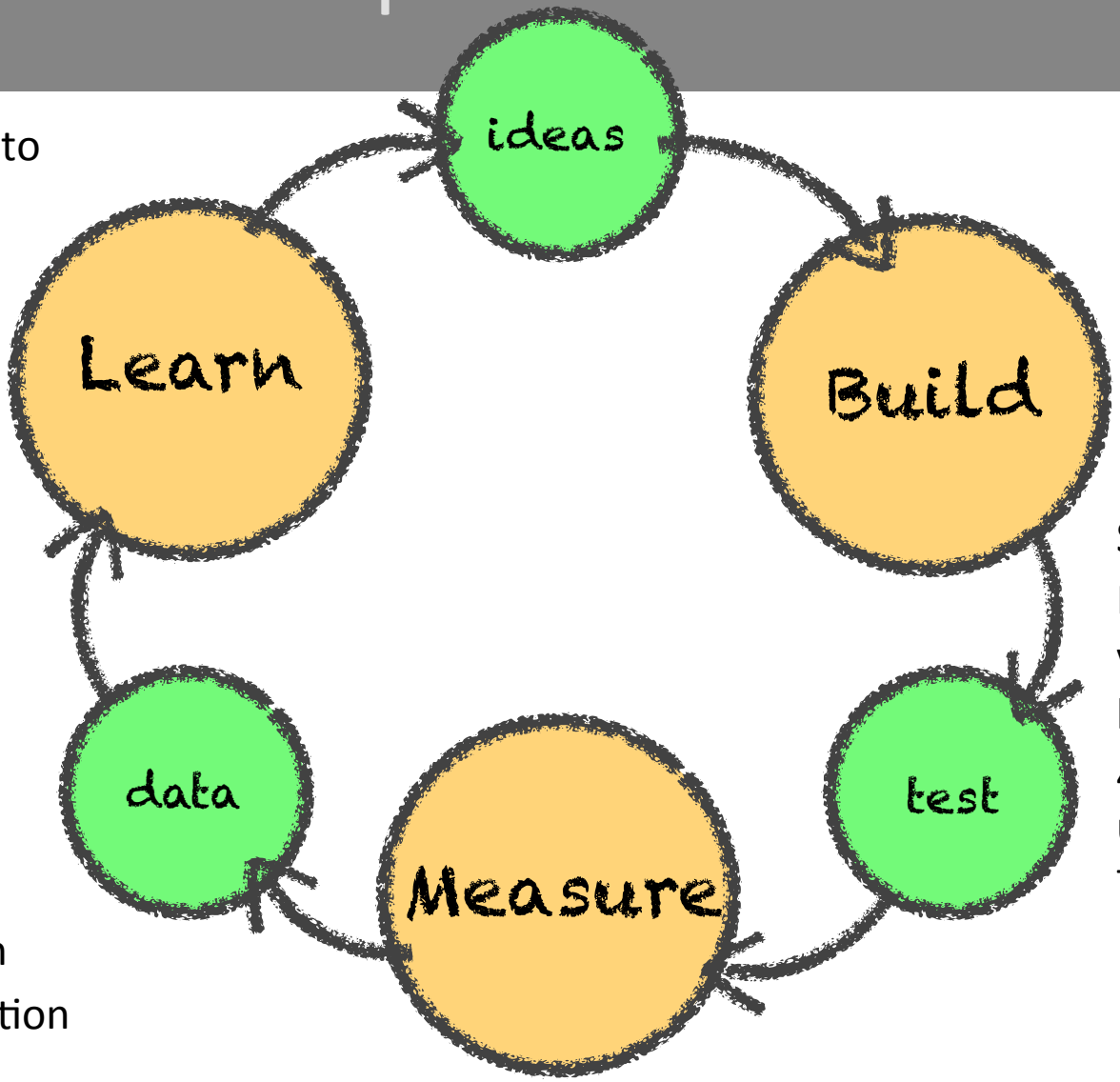
(cc) BY-NC-SA

The Lean Startup Validated Learning Loop

Simple models to distill findings

Ideation

Prototype & evaluate inside the team



Simple prototype

Live data prototype

Value test (like a landing page)

404 Test (like unimplemented features)

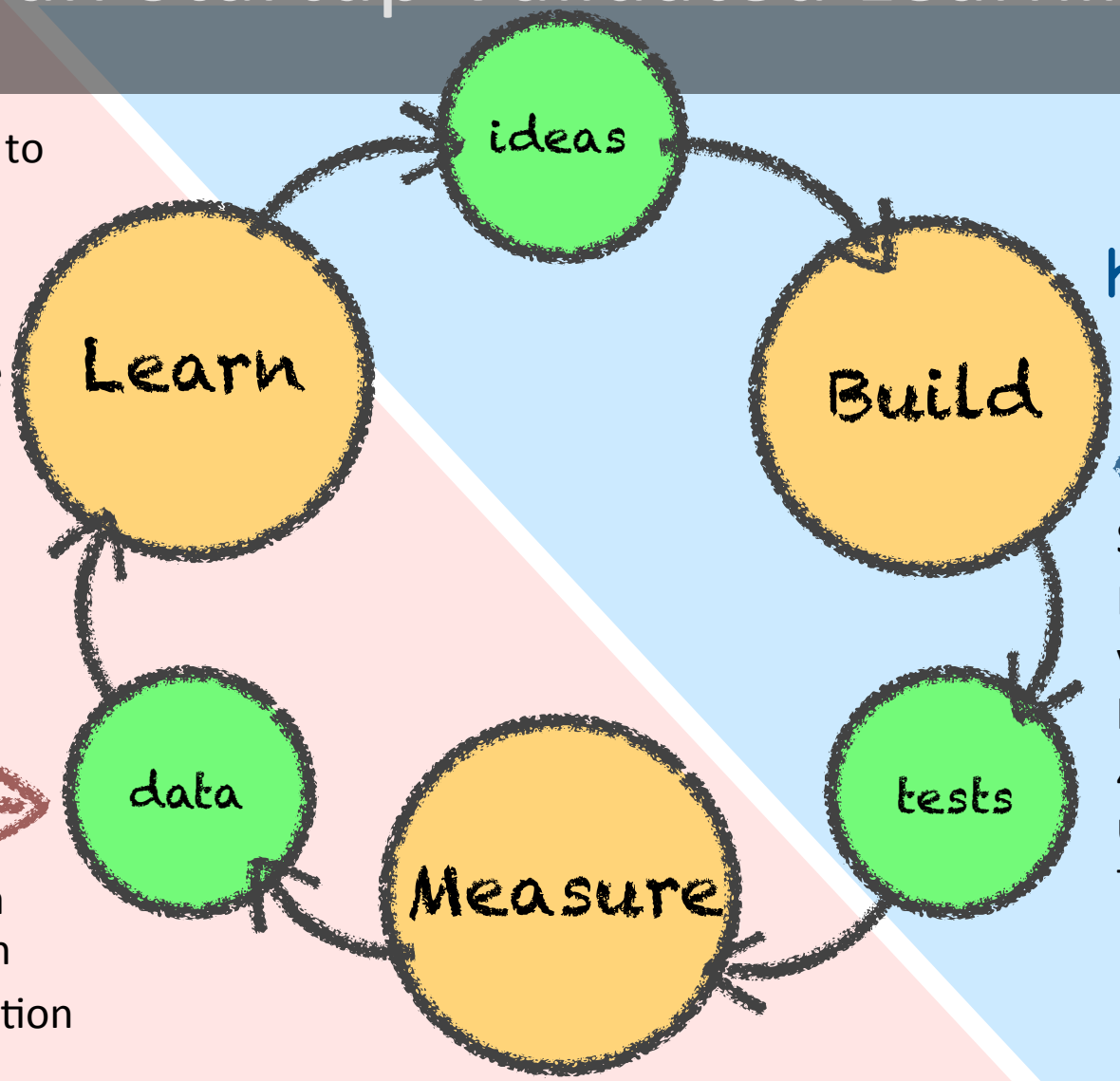
Metrics from a live application

Direct observation

Poll or survey

(cc) BY-NC-SA

The Lean Startup Validated Learning Loop



Simple models to distill findings

Ideation

Prototype & evaluate inside the team

happy, hopeful, hypothetical stuff

Simple prototype

Live data prototype

Value test (like a landing page)

404 Test (like unimplemented features)

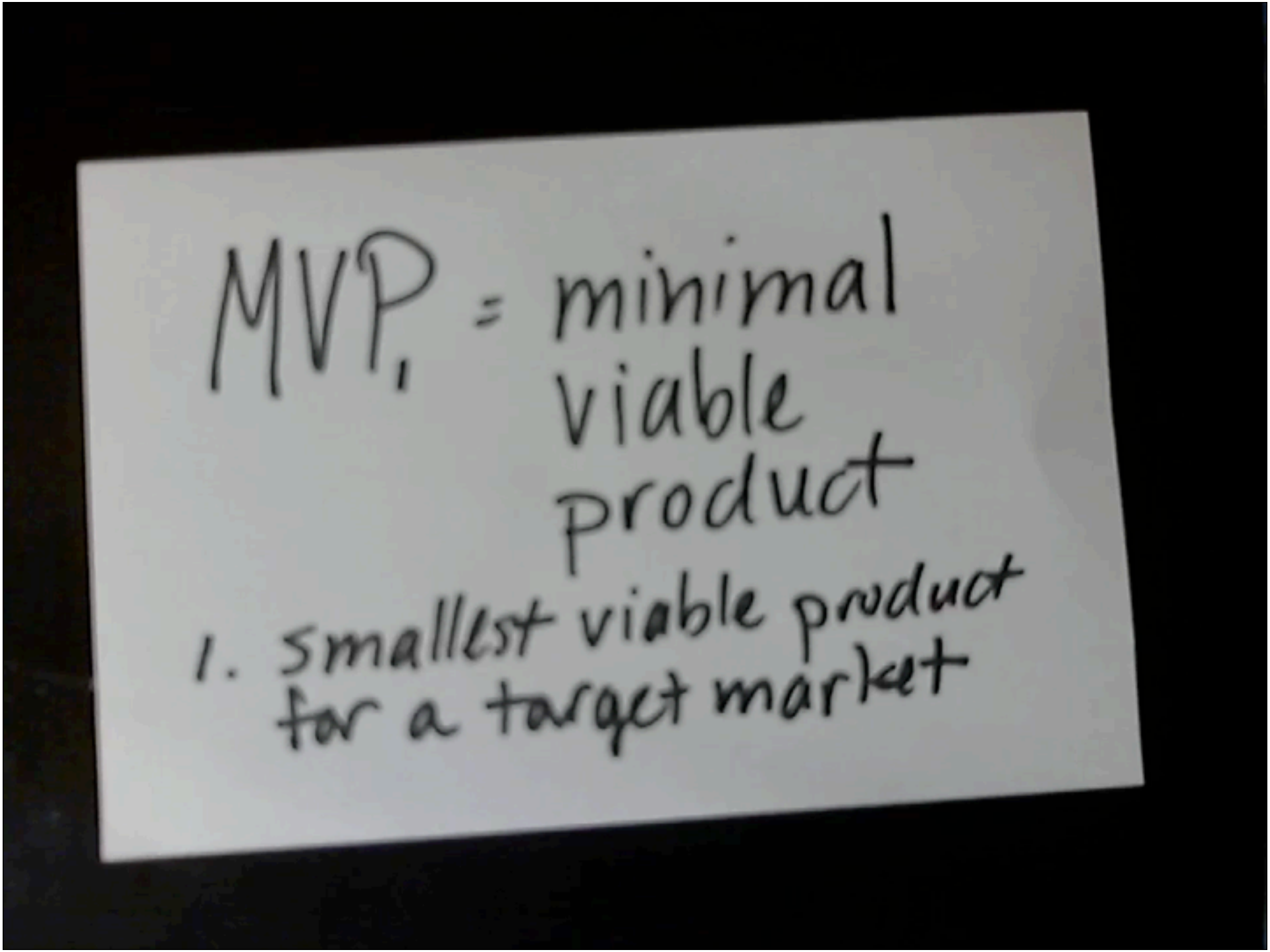
harsh reality

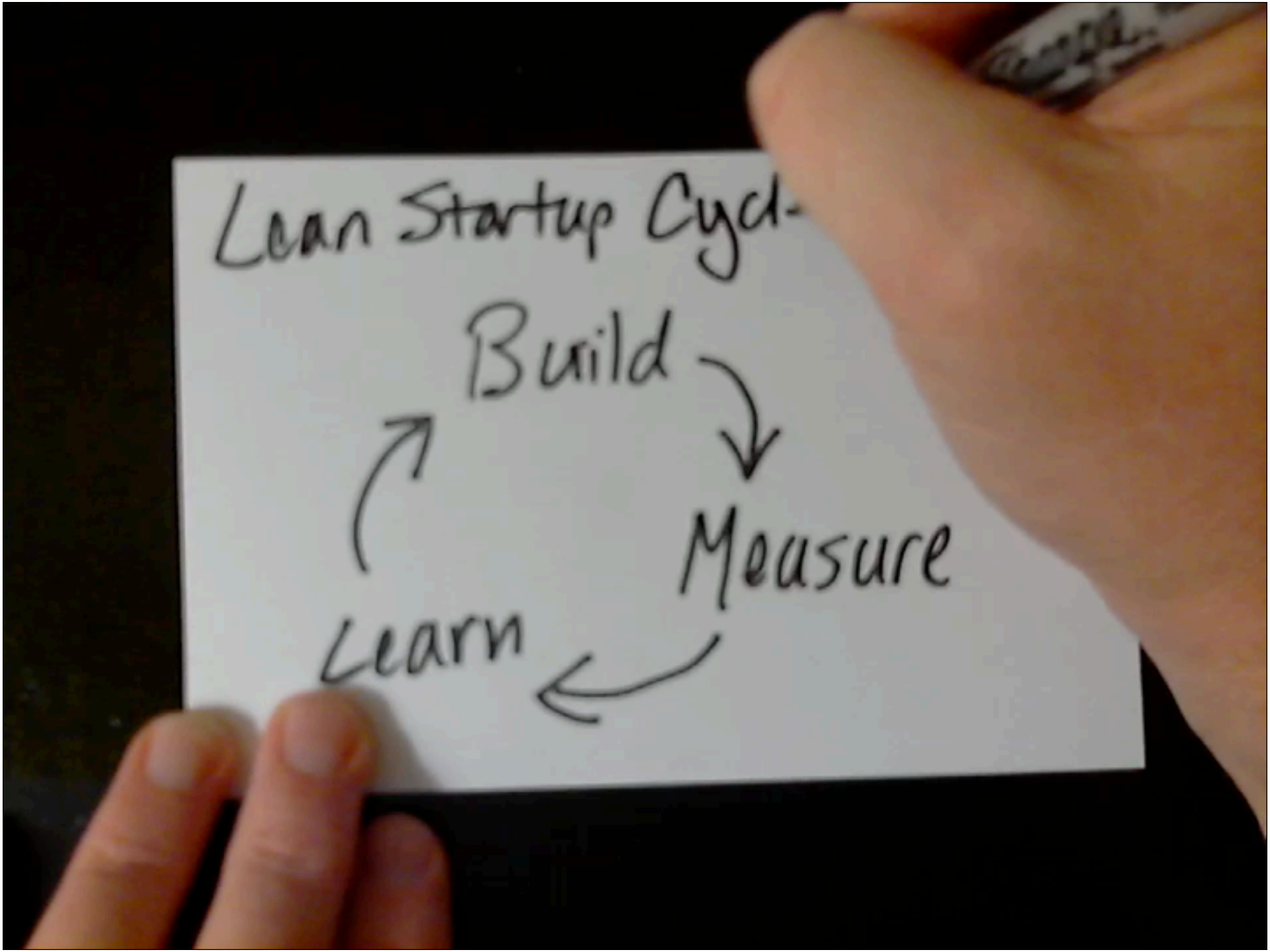
Metrics from a live application

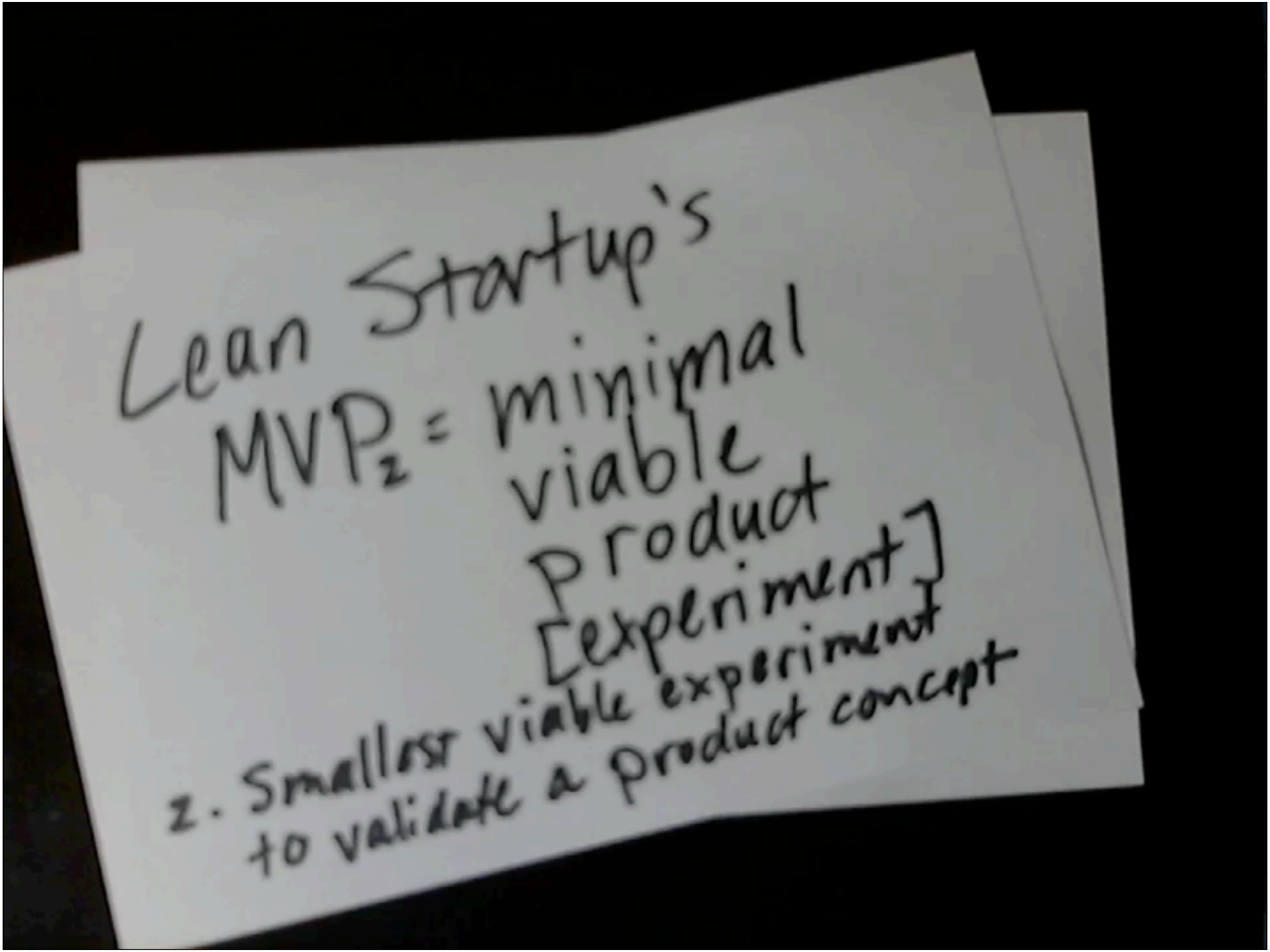
Direct observation

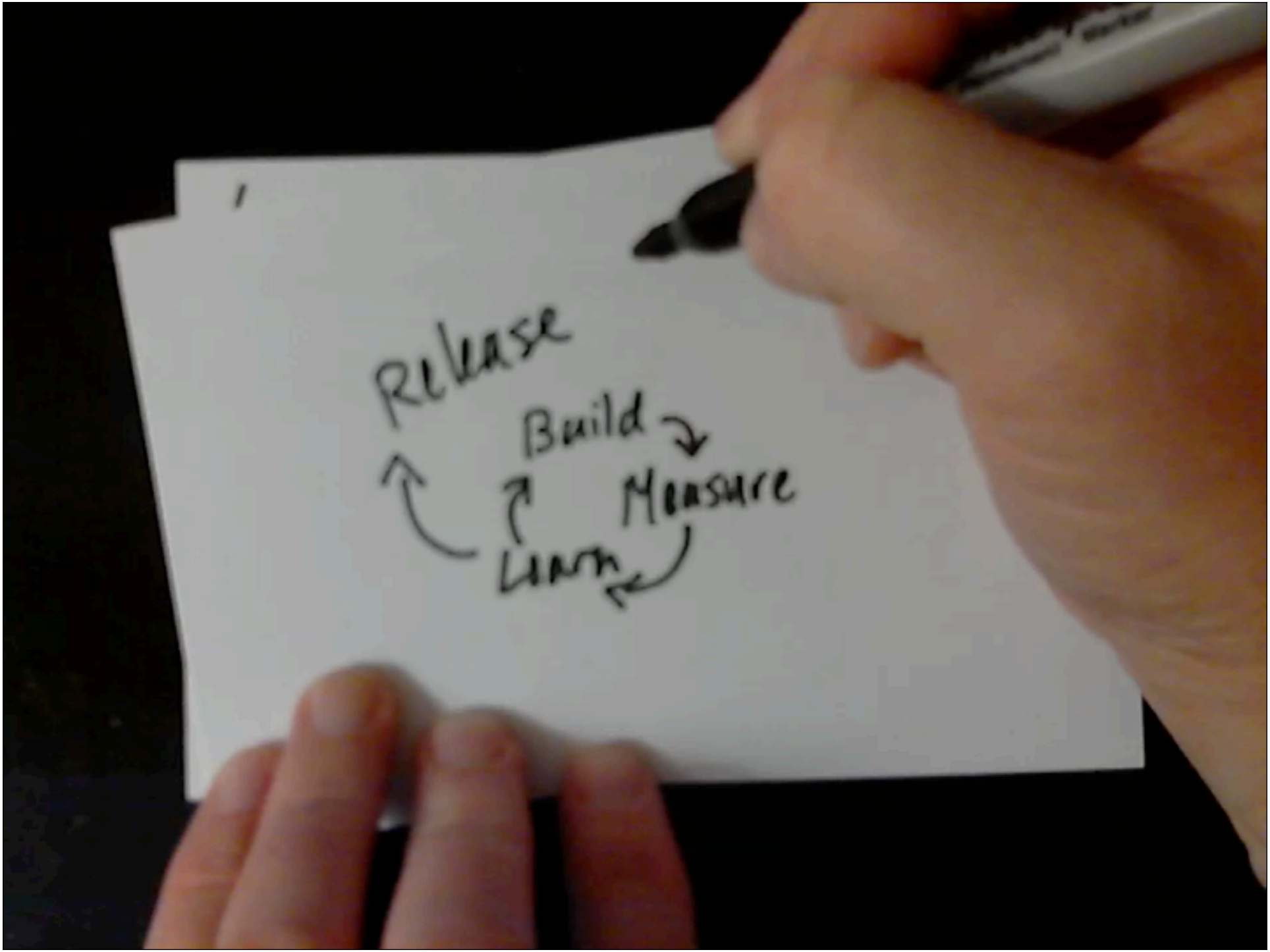
Poll or survey

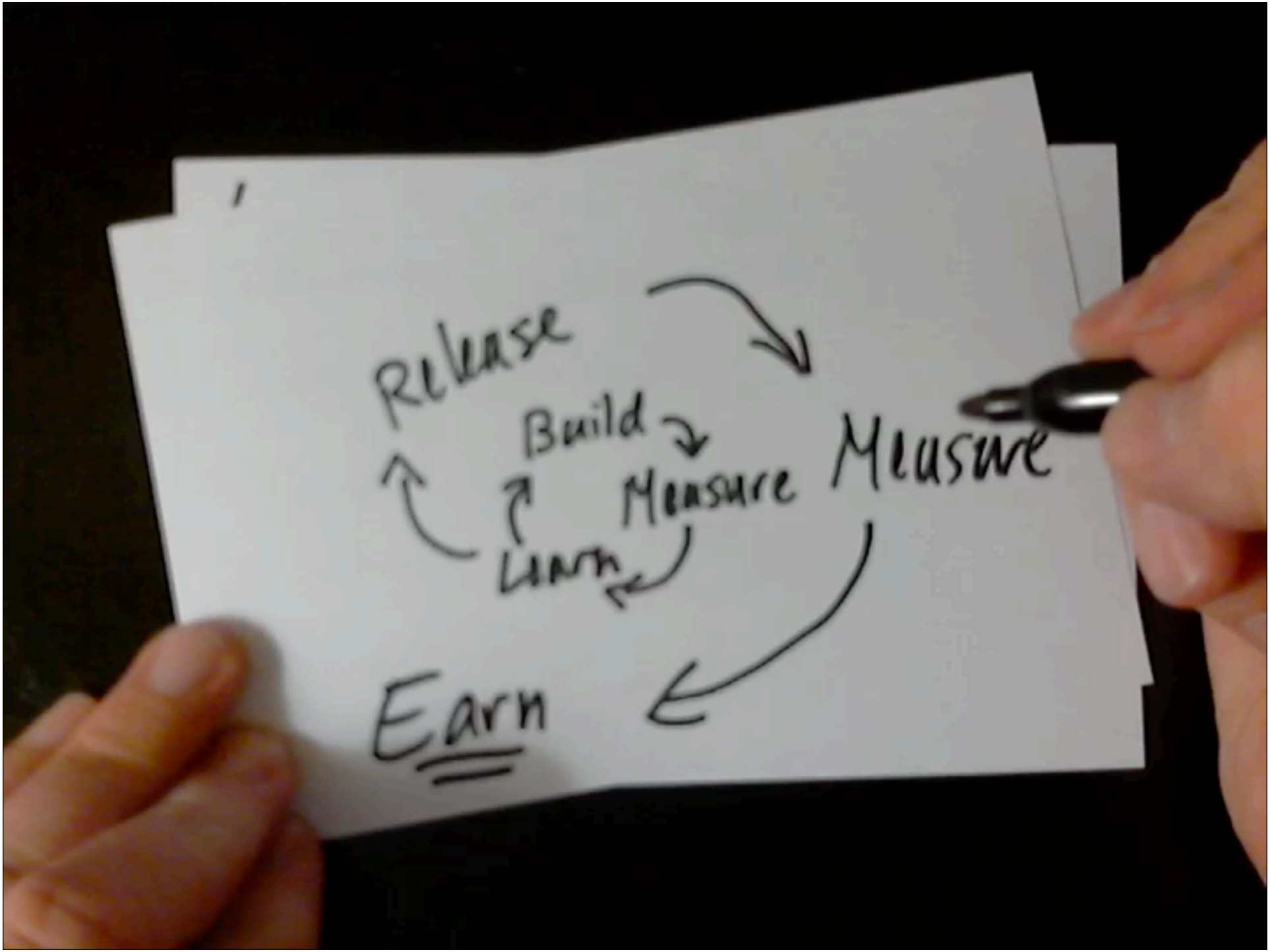
(cc) BY-NC-SA

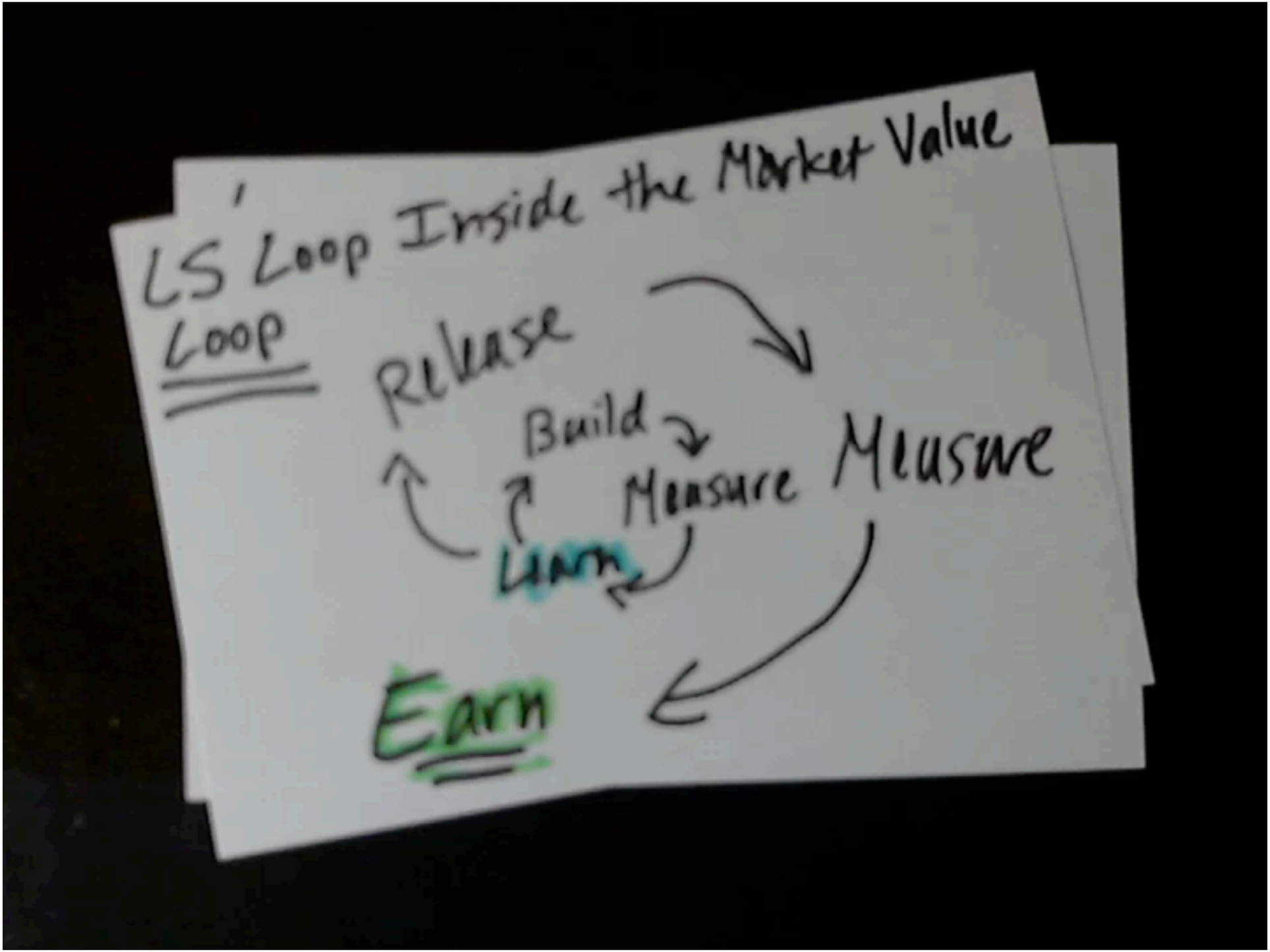

MVP = minimal viable product
1. smallest viable product for a target market

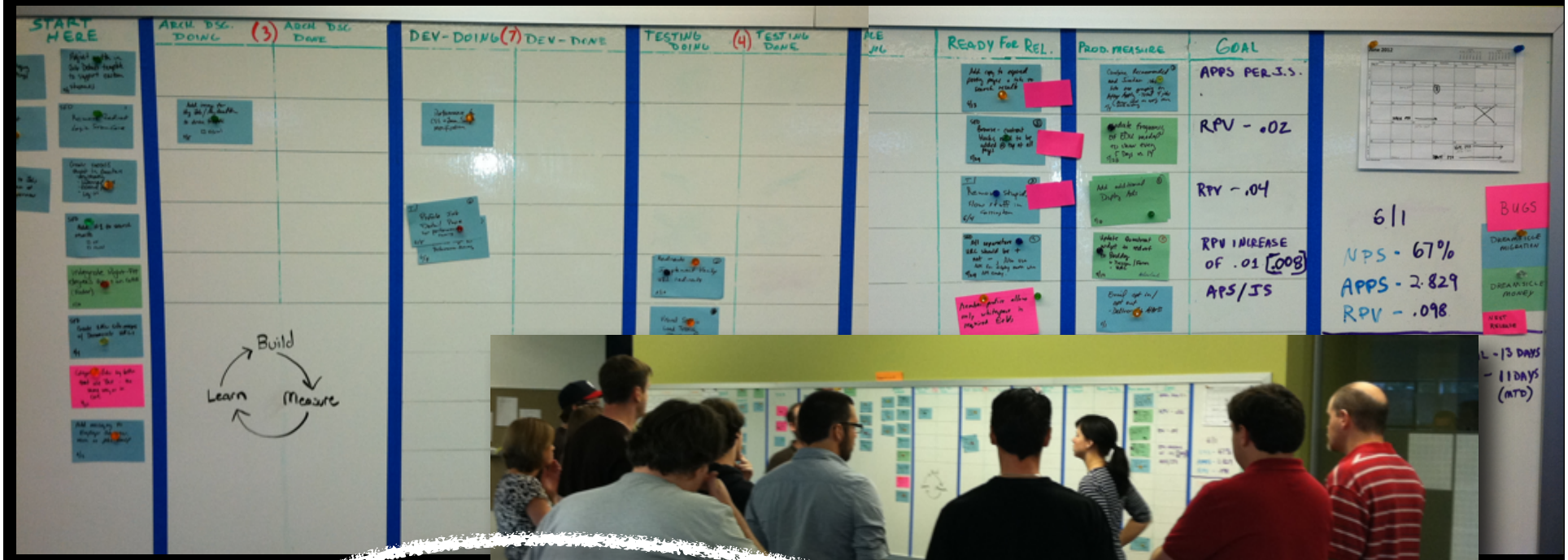

Lean Startup Cycl-
Build
Measure
Learn

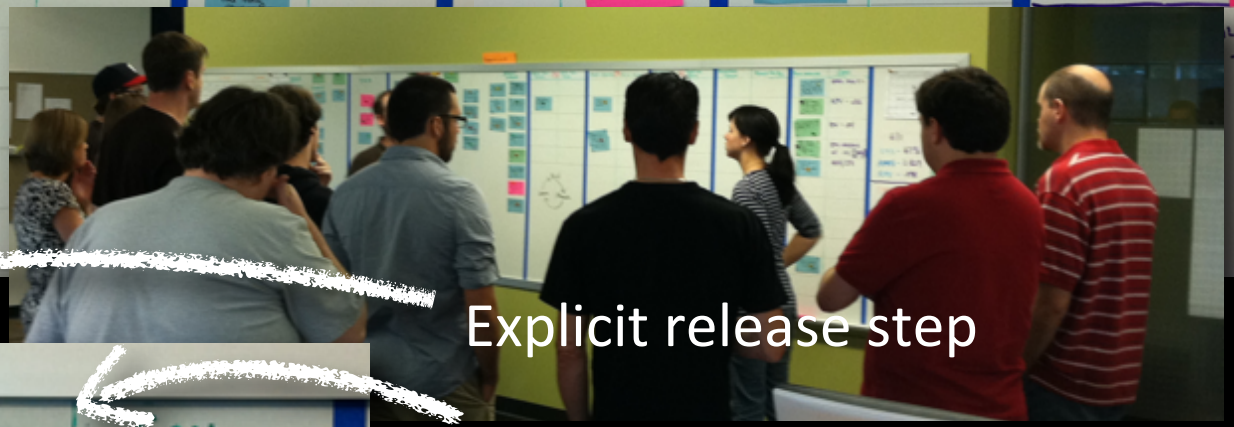

Lean Startup's
MVP2 = minimal
viable
product
[experiment]
2. Smallest viable experiment
to validate a product concept


Release
Build
Measure
Learn

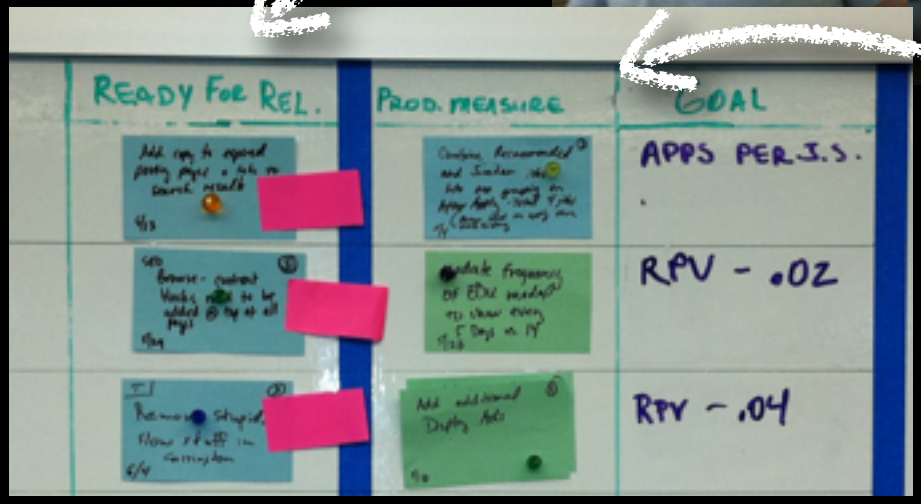

Release
Build
Measure
Learn
Measure
Earn


LS Loop Inside the Market Value Loop
Release
Build
Measure
Learn
Measure
Earn





Explicit release step



Explicit measure step & metrics

Nothing leaves the board until there's been a discussion on what we've learned

Snag-a-Job's board courtesy of David Bittenbender

(cc) BY-NC-SA

Nordstrom Innovation Lab's Learning Loop

http://www.startuplessonslearned.com/2011/10/case-study-nordstrom-innovation-lab.html

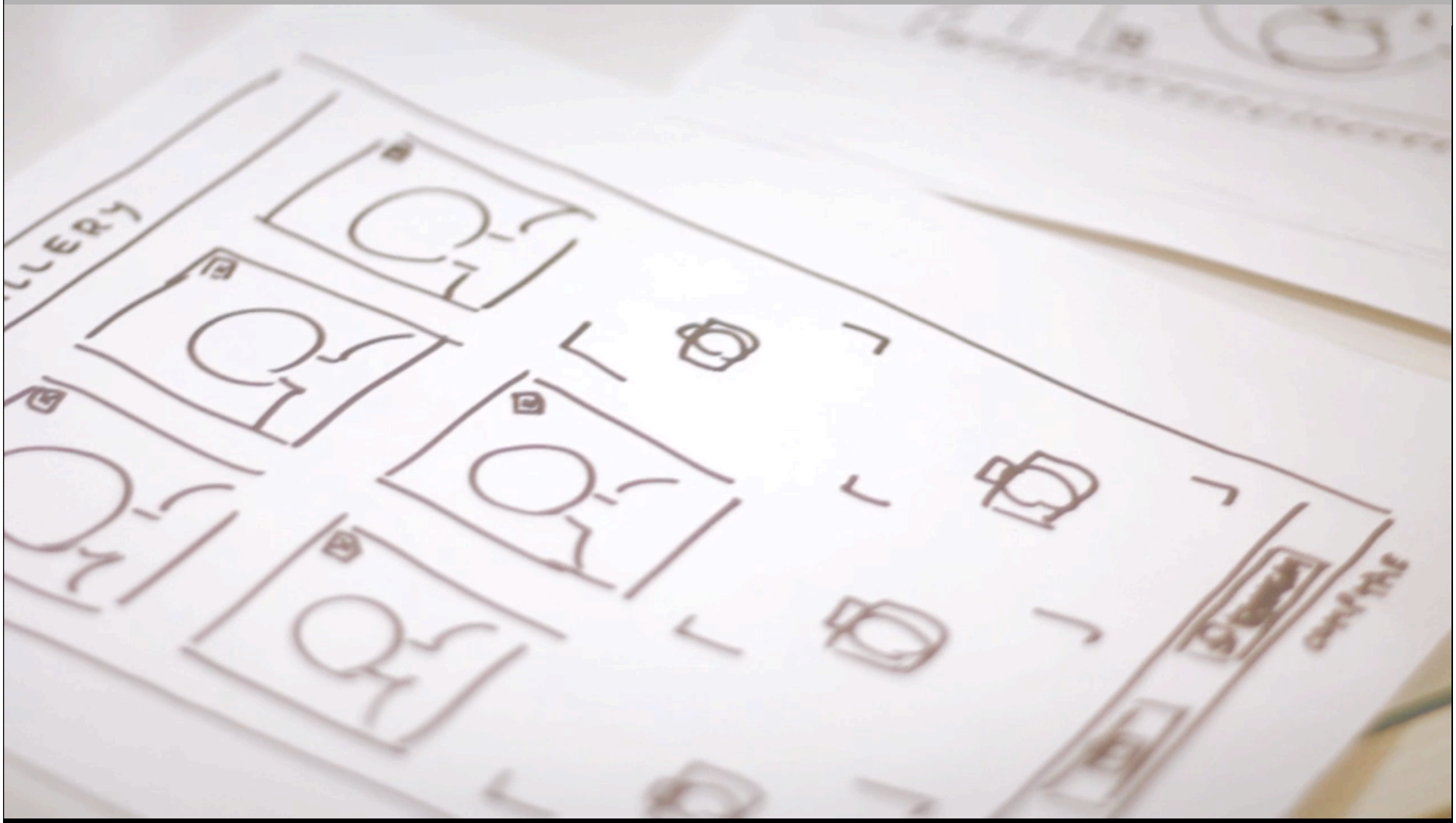
(cc) BY-NC-SA

Nordstrom Innovation Lab's Learning Loop

http://www.startuplessonslearned.com/2011/10/case-study-nordstrom-innovation-lab.html

(cc) BY-NC-SA

Nordstrom Innovation Lab's Learning Loop



http://www.startuplessonslearned.com/2011/10/case-study-nordstrom-innovation-lab.html

(cc) BY-NC-SA

Nordstrom Innovation Lab's Learning Loop

http://www.startuplessonslearned.com/2011/10/case-study-nordstrom-innovation-lab.html

(cc) BY-NC-SA

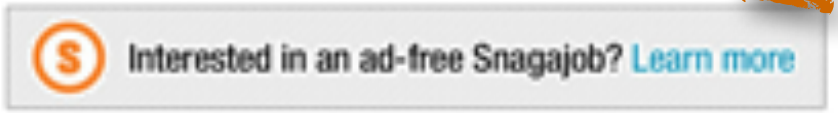
Nordstrom Innovation Lab's Learning Loop

http://www.startuplessonslearned.com/2011/10/case-study-nordstrom-innovation-lab.html

(cc) BY-NC-SA

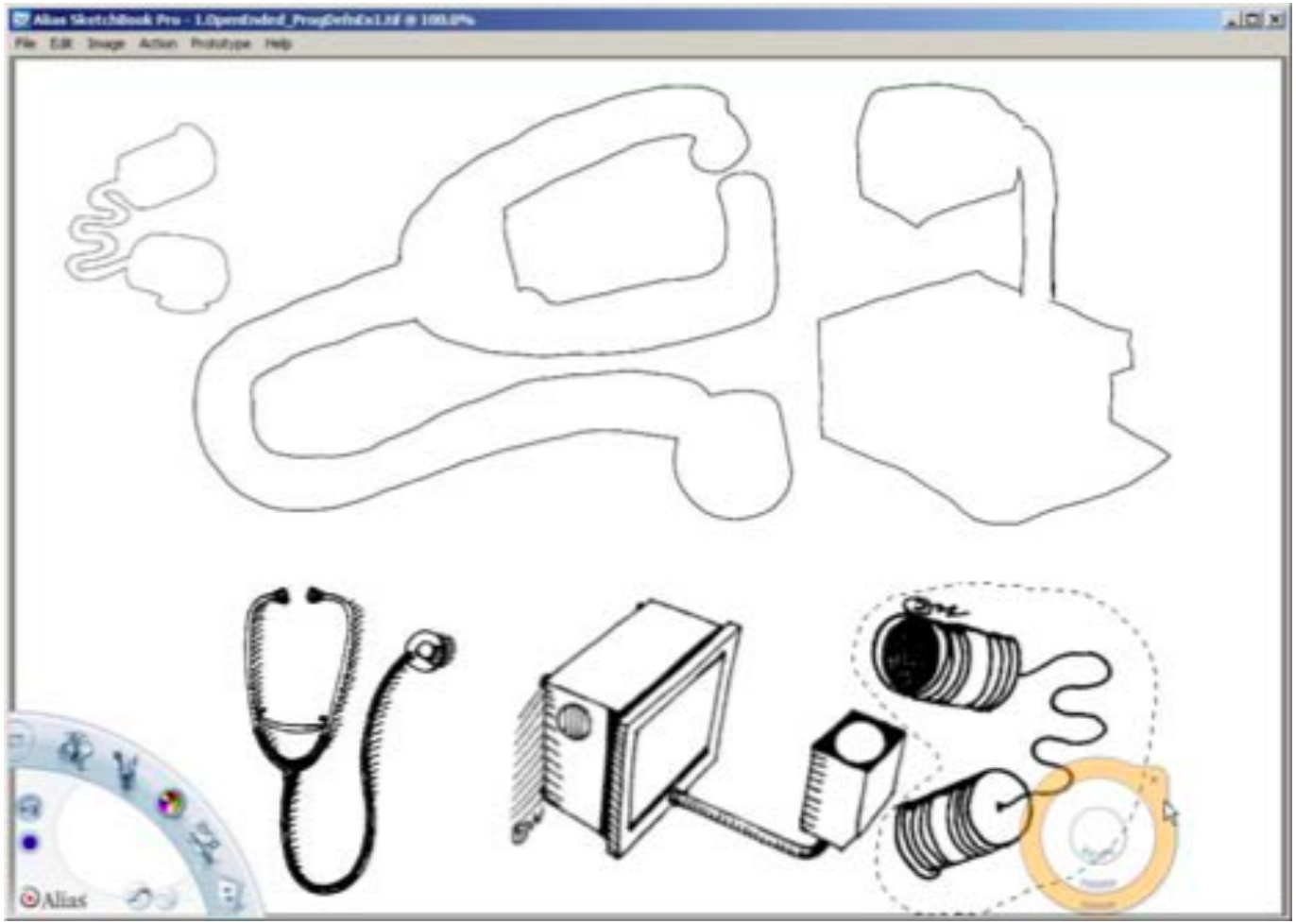
SnagaJob uses experiments in their live site to learn

We believe all these ads turn people off.



SnagAJob uses "404 Tests" or "buttons to nowhere."

Sketchbook Pro's Minimal Viable Product Experiment for rotate and resize



(cc) BY-NC-SA

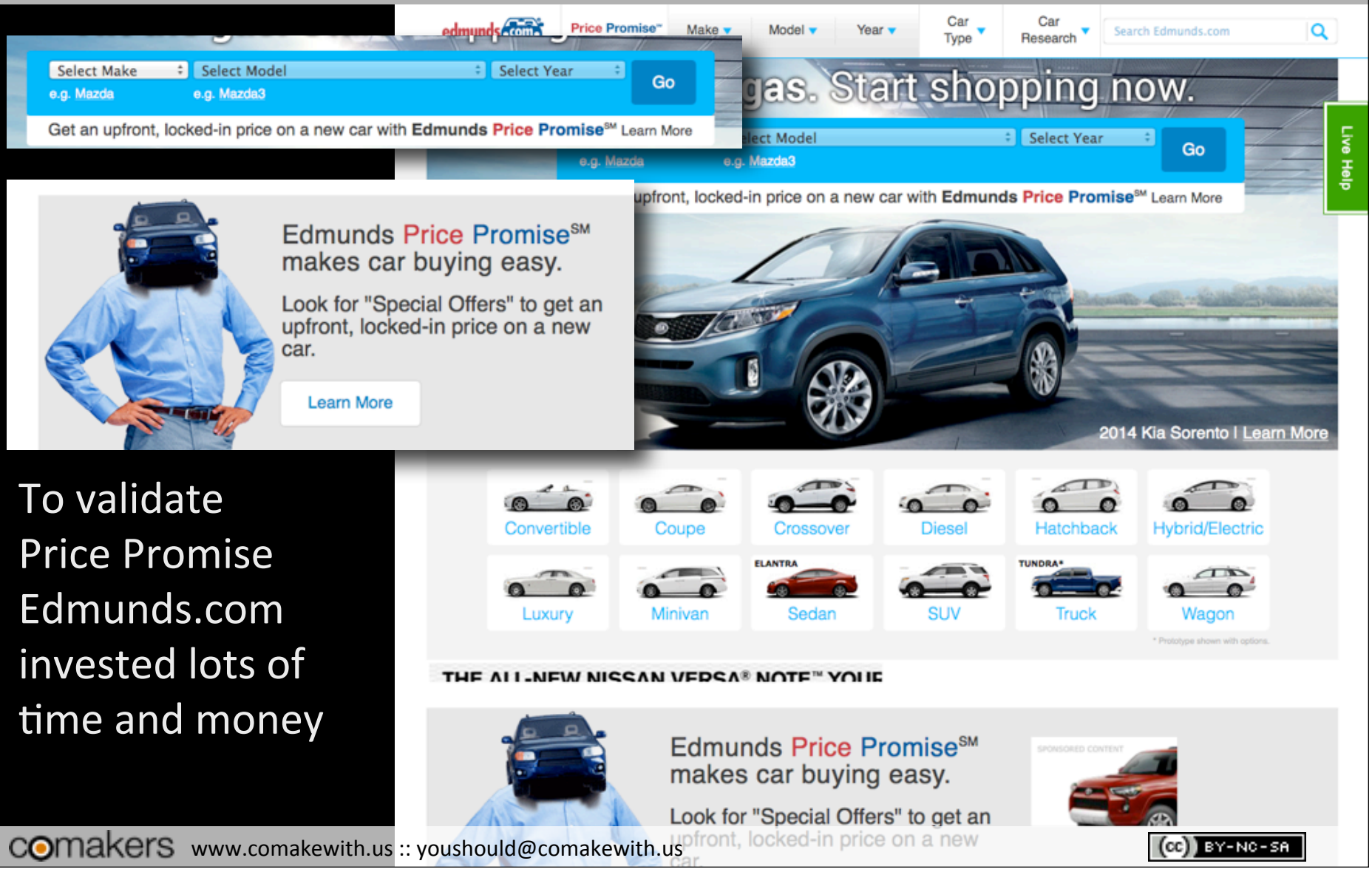
Adding Design Thinking to fix Lean Startup

Nordstrom recognizes something is missing...

(cc) BY-NC-SA

Adding Lean Startup to fix Design Thinking

It’s easy to fool yourself



To validate Price Promise Edmunds.com invested lots of time and money

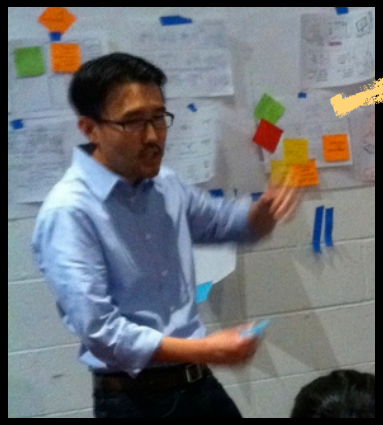
(cc) BY-NC-SA

We've gotten a lot
better at measuring outcomes

Before, we
would have spent a lot of
time arguing about what to do.
We wouldn't have spent
money and time to learn.
Now we do.

"We had the misperception that the process, the practice, or the methods will automatically produce success.

Occasionally that's the case, but most of the time it isn't"



(cc) BY-NC-SA

The fastest way to failure is to believe you're safe

(cc) BY-NC-SA

Recognize process as a game strategy and play to win

Invest in understanding the problems you're solving

Invest in validating your solutions

Minimize output, maximize outcome

(cc) BY-NC-SA

SAFETY NOT GUARANTEED

How great teams break the rules to create successful products

Jeff Patton
Chief troublemaker, Comakers LLC
jeff@comakewith.us, twitter: @jeffpatton