

Using Design Thinking to Stop Building Worthless Software



Jeff Patton


jpatton@acm.org

twitter: @jeffpatton


www.agileproductdesign.com



Process is a bit of a racket

A man in a dark suit, light-colored striped shirt, and dark tie. He has a serious, angry expression and is holding a yellow banana in his right hand.

Hello... our customers
are complaining because they
hate our software!


A man in a dark suit, white shirt, and red tie. He is smiling broadly and looking towards the camera.

We believe if we could
ship more of it faster, that would
solve our problems!

I've got a process
that'll fix everything!

Process always seems our focus on
speed of delivery

Return on investment from
software is far from
certain



Typically about 50%
to 80% of all software we
ship fails to accomplish it's
objectives.

INSPIRED

HOW
TO
CREATE
PRODUCTS
CUSTOMERS
LOVE

People like Marty say this stuff is hard

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

Source: Stevens, G.A. and Burley, J., “3,000 Raw Ideas = 1 Commercial Success!”,
(May/June 1997) Research Technology Management, Vol. 40, #3, pp. 16-27.



For every 4 projects that enter development, only 1 makes it to the market

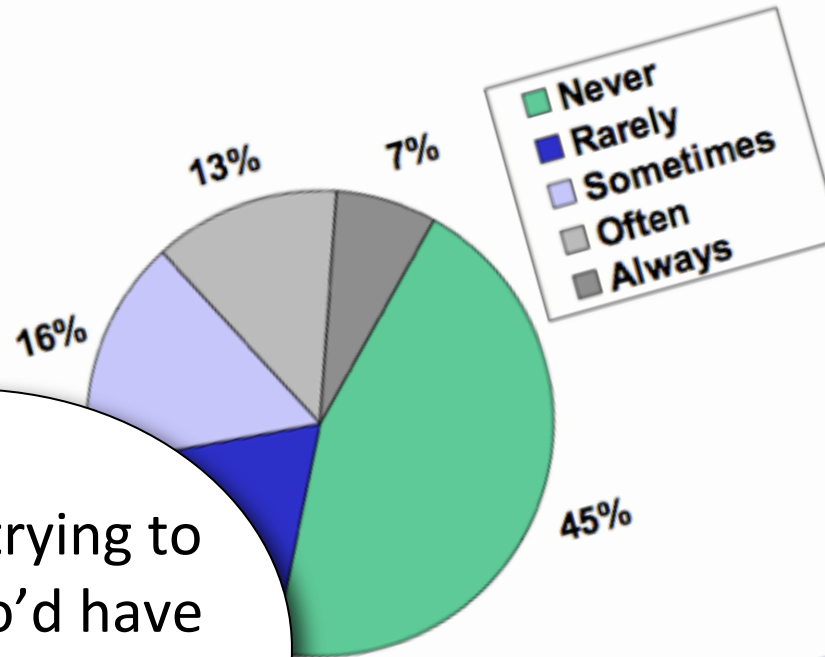
At launch, at least 1 of 3 products fail despite research and planning

An estimated 46% of all resources allocated to product development and commercialization by U.S. firms is spent on products that are cancelled or fail to yield an adequate financial return.

Source : Winning at New Products, p.9

Features & Function Usage

THE
STANDISH
GROUP



I was just trying to help... who'd have thought it'd turn out so bad.

According to the Standish Group, 64% of features and functions are rarely or never used

Copyright © 2002 The Standish Group International, Inc.

It's only after delivery that we understand value

opportunity:
integrated music
management and
portable music
player

high
demand, high
value solution:



low
demand, low
value solution:



“There were plenty of weak spots that led to Microsoft's disastrous December quarter, but one that didn't get much attention Thursday was how badly the Zune did.”

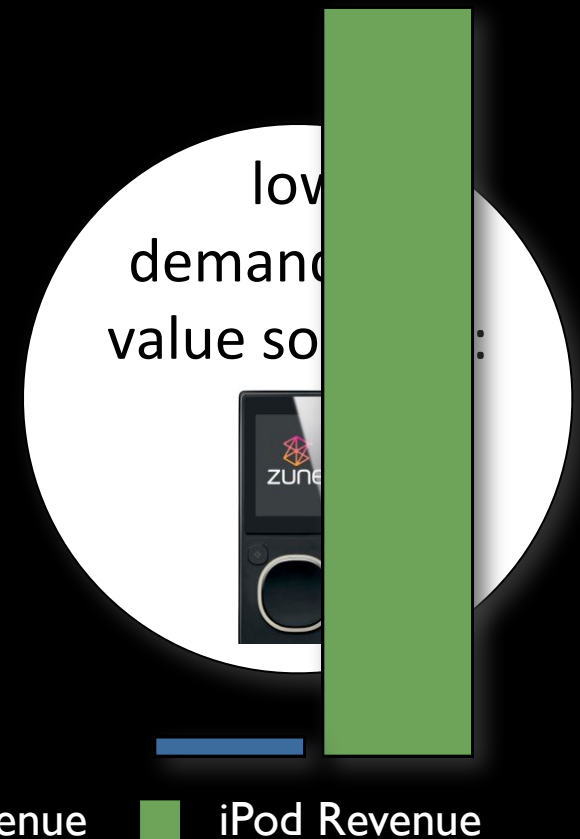
--Ina Fried, CNet News,
January 2009

What's the business value for the same feature in Apple's and Microsoft's backlog?



"Zune platform revenue decreased \$100 million, or 54 percent, reflecting a decrease in device sales,' Microsoft said. That's quite a drop."

Apple, by contrast, saw its iPod unit sales up 3 percent, while revenue dropped by 16 percent. It still racked up \$3.3 billion in revenue, as compared with less than \$100 million for the Zune."



■ Zune Revenue ■ iPod Revenue



Prioritize by business value?

Is that really your best advice?

Delivering stupid software
faster just get's you more
stupid software

Edmunds.com gets it



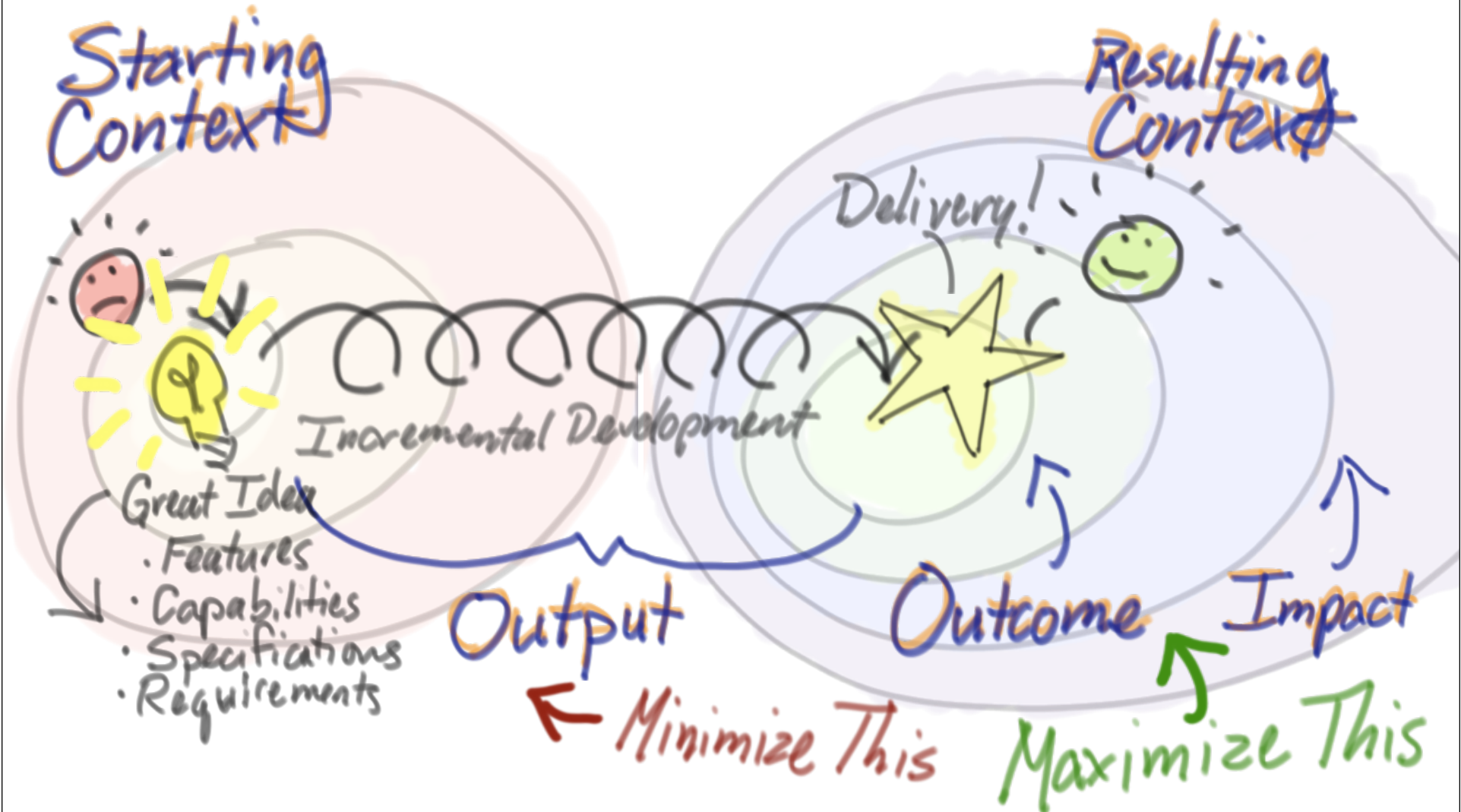
A screenshot of the Edmunds.com website interface. At the top left is the Edmunds.com logo. To the right, there is a navigation bar with links for "Welcome, Guest", "Sign In", "Facebook Sign In", "Account", "Mobile", "Help", "Newsletter", and "Inside Line". Below this is a secondary navigation bar with categories: "NEW CARS", "USED CARS", "CAR REVIEWS", "TIPS & ADVICE", "MAINTENANCE", and "FORUMS". A search bar with the text "Search Site" and a "FIND Q" button is on the right. The main content area is divided into two columns. The left column features a "Find your next car" section with a "Zip Code: 80022" field and an "UPDATE" button. Below this is a "Search by Type" section with four car icons: SUV, Sedan, Truck, and Minivan. Further down are sections for "Get FREE Price Quotes" and "Search Used Cars in Your Area" powered by AutoTrader. The right column is an "ADVERTISEMENT" section titled "New Vehicle Spotlight" featuring a blue 2011 Honda Accord and the text "2011 Honda Accord".

the law of indirect consequences

We're not building software,
we're changing the world

(yes, I know that sounds cheesy)

Place focus on how the world will change after shipping software



output



we build this

outcome



we want this

Design thinking is “in style”

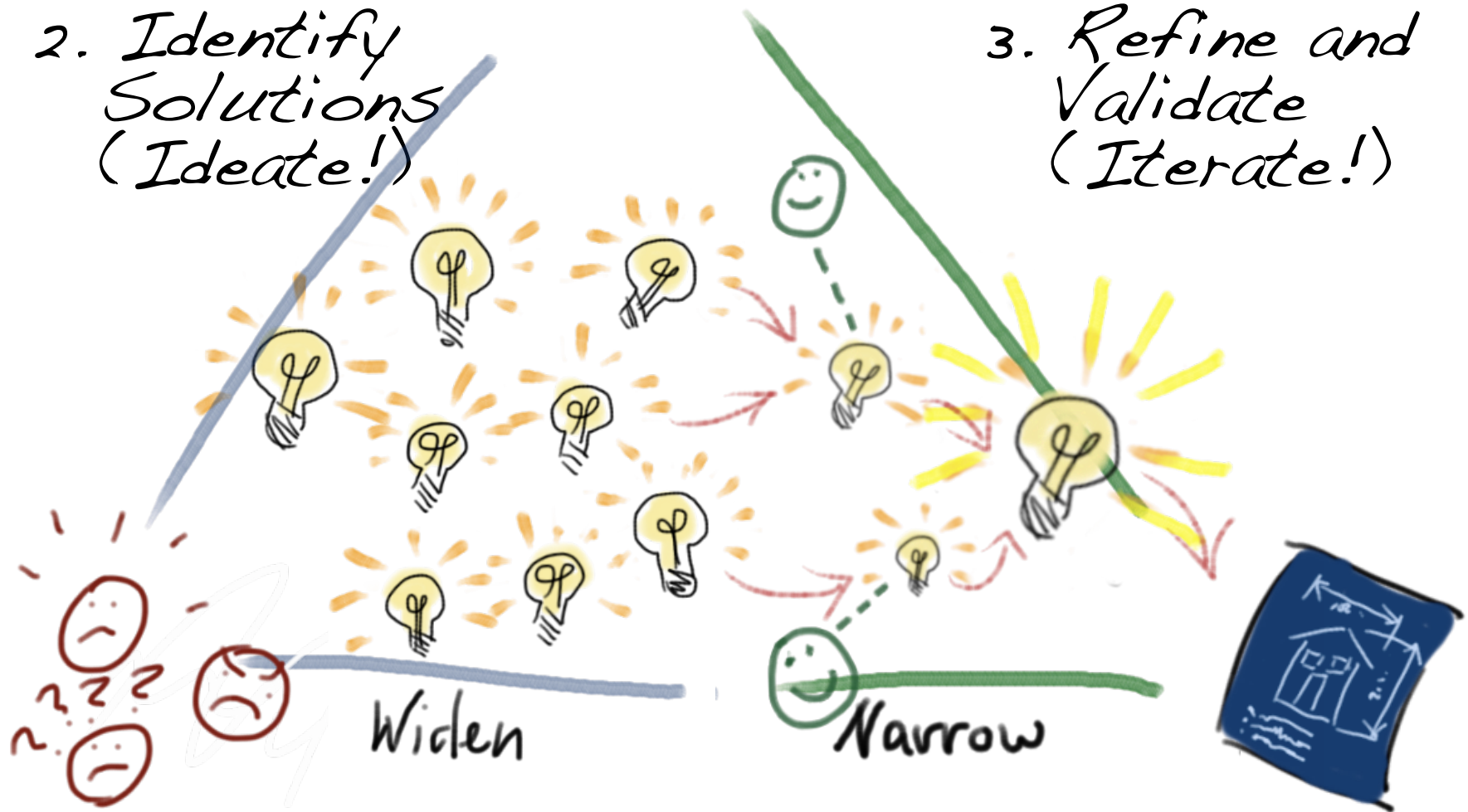


It was Matt and his advice that kicked off Edmunds.com's quest

Design thinking follows a simple process

2. Identify Solutions
(Ideate!)

3. Refine and Validate
(Iterate!)



1. Understand the problem

4. Create a plan

Research + analysis
makes sense of the
world as it is

Design imagines the
world differently than
it is today

* From Lawson's How Designers Think

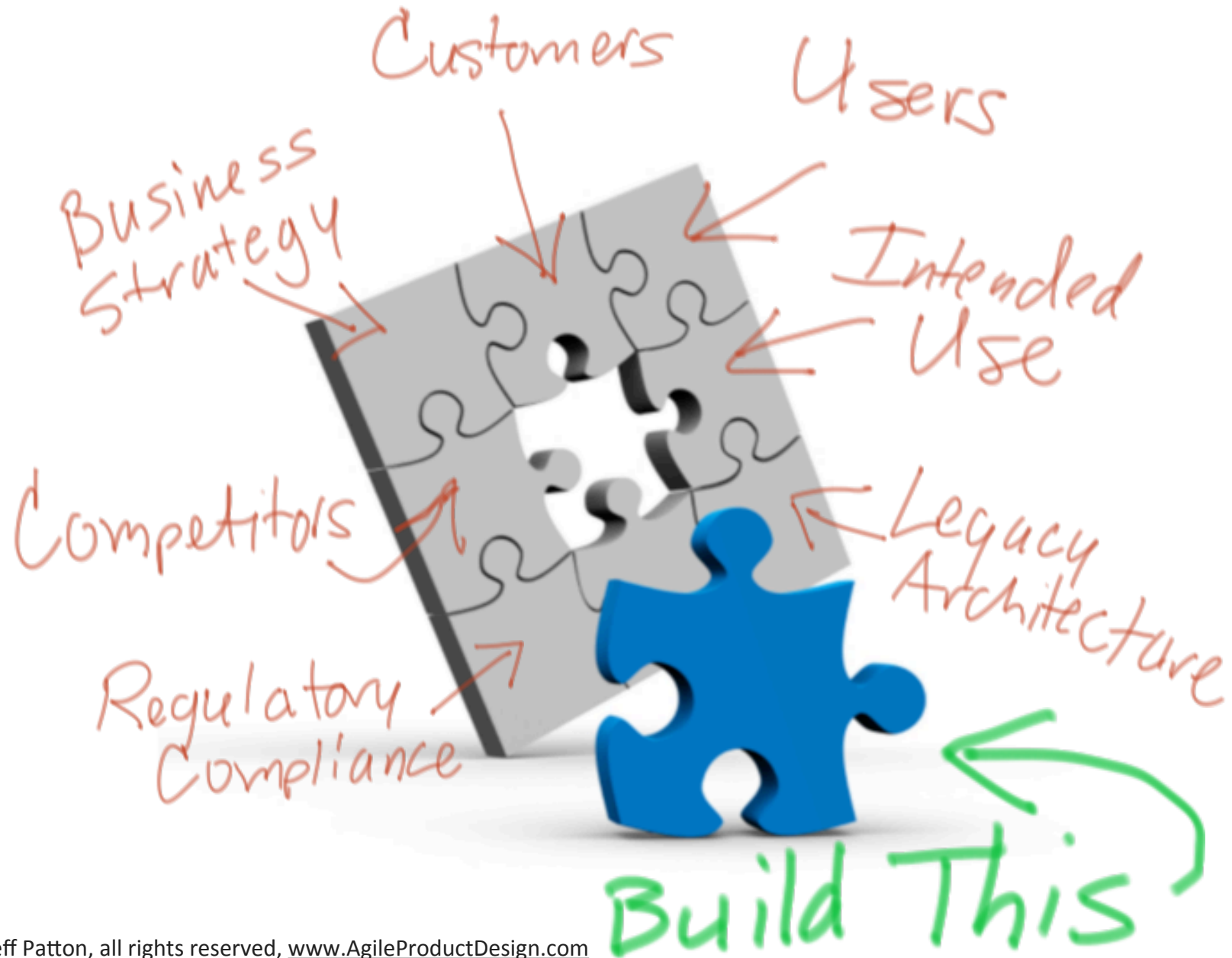
Humans are notoriously bad at predicting the future



1

Understand context and
define the problem

Software fits a problem-shaped hole



Genchi Genbutsu

Learning, not capturing requirements



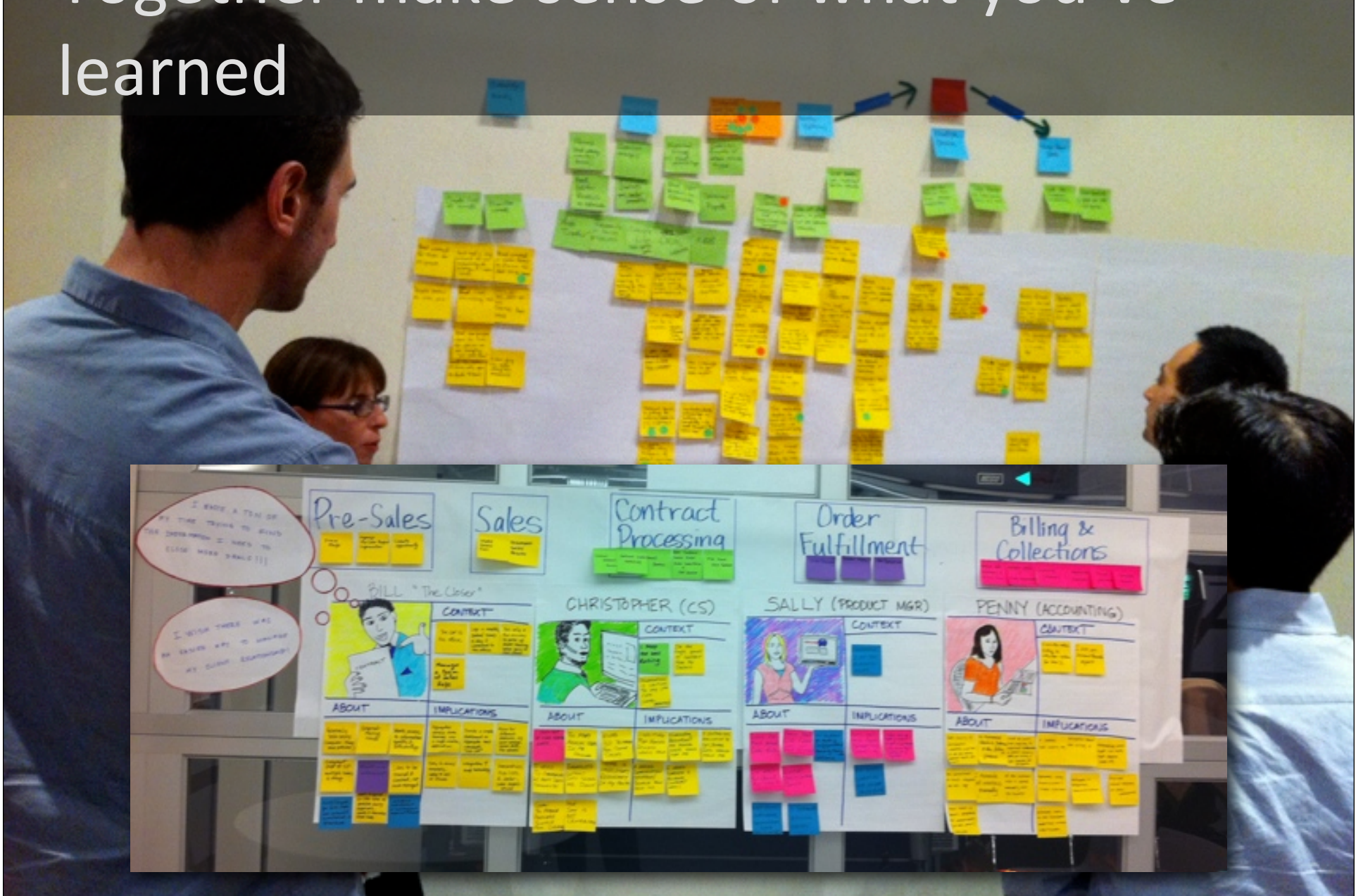
Ask your users to tell their stories...
they're dying to



Let your users teach you their job



Together make sense of what you've learned



Ad hoc models and maps get us on the same page

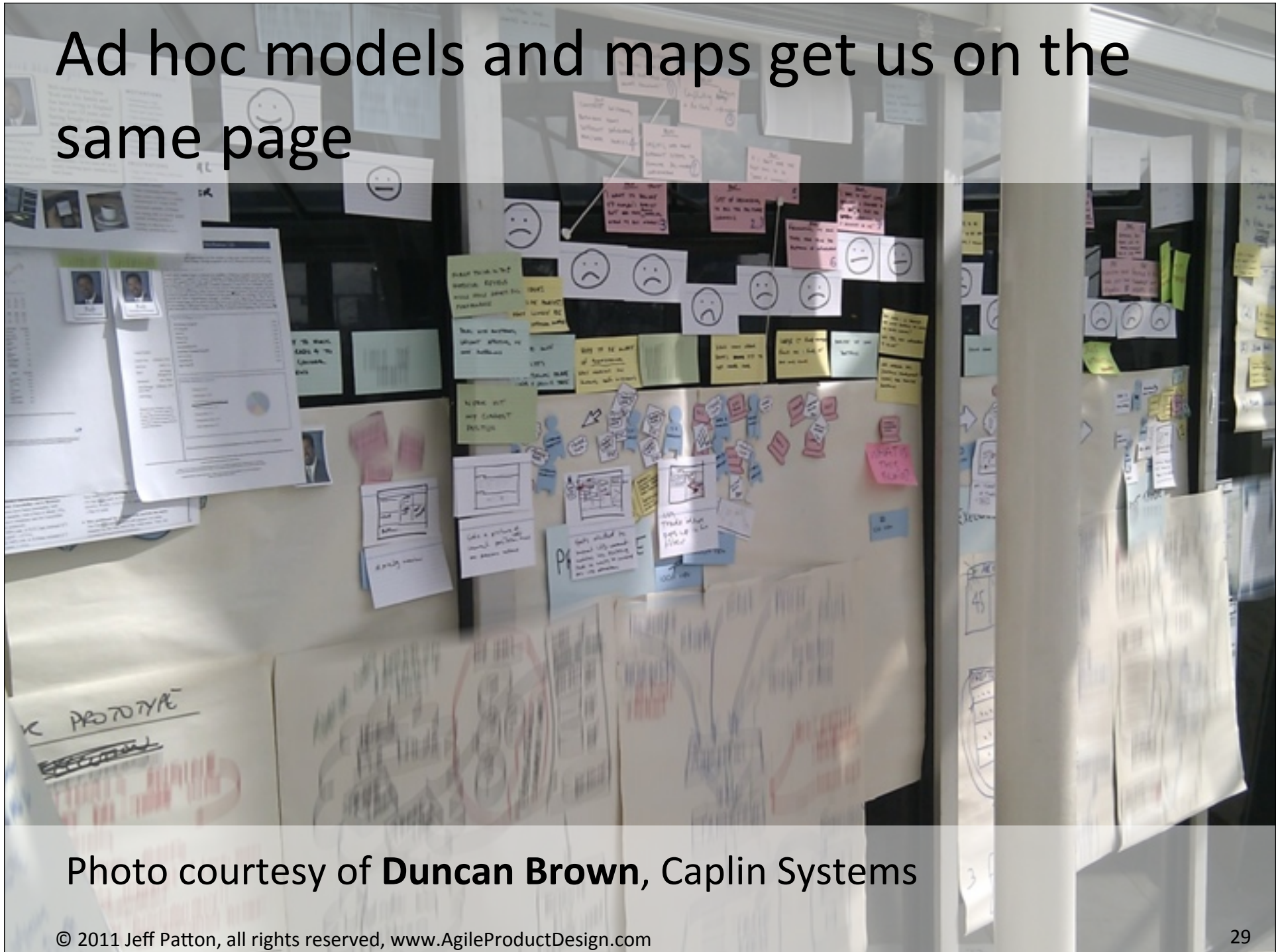
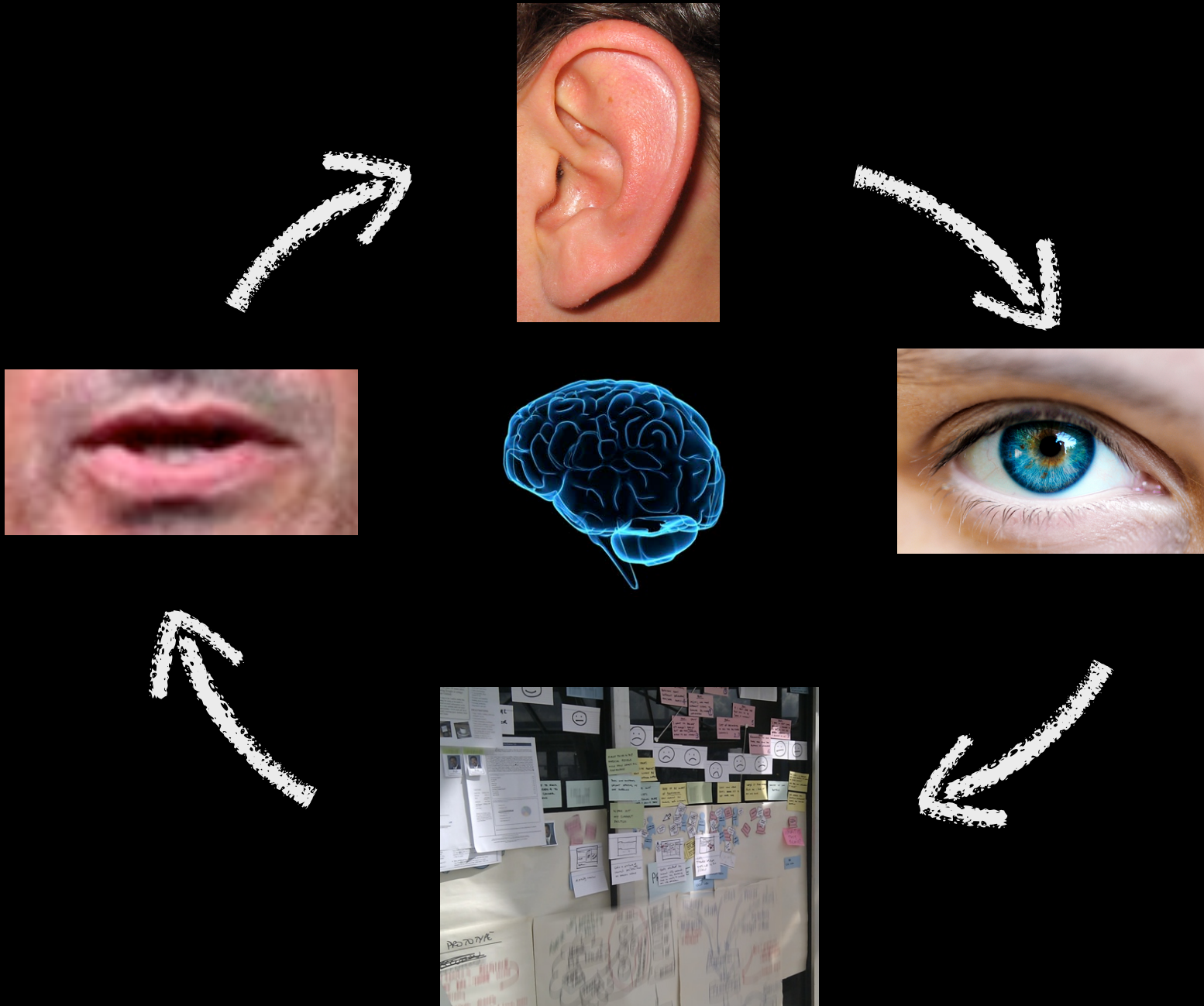


Photo courtesy of **Duncan Brown**, Caplin Systems



When we talk about ideas, we often believe we understand each other



"I'm glad we all agree then"

Representing our ideas as models allows us to detect inconsistencies



Through discussion and iterative model building we arrive at a stronger shared understanding



We can leverage the shared understanding to problem solve together



"I'm glad we all agree then"

Choose the problems in your context you'd like to address



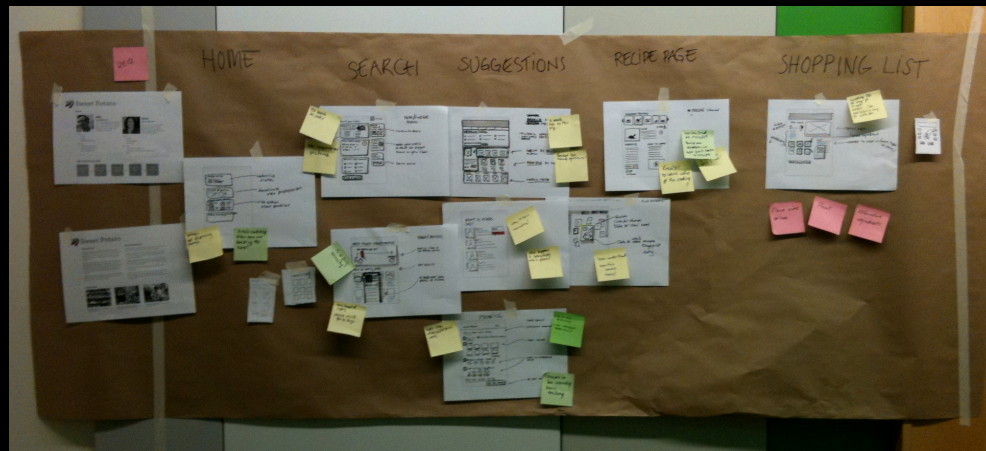


Ideate to identify
solutions

Circles



A design studio approach engages the whole team in sketching



Sketchboarding is collaborative way to come up with UI ideas



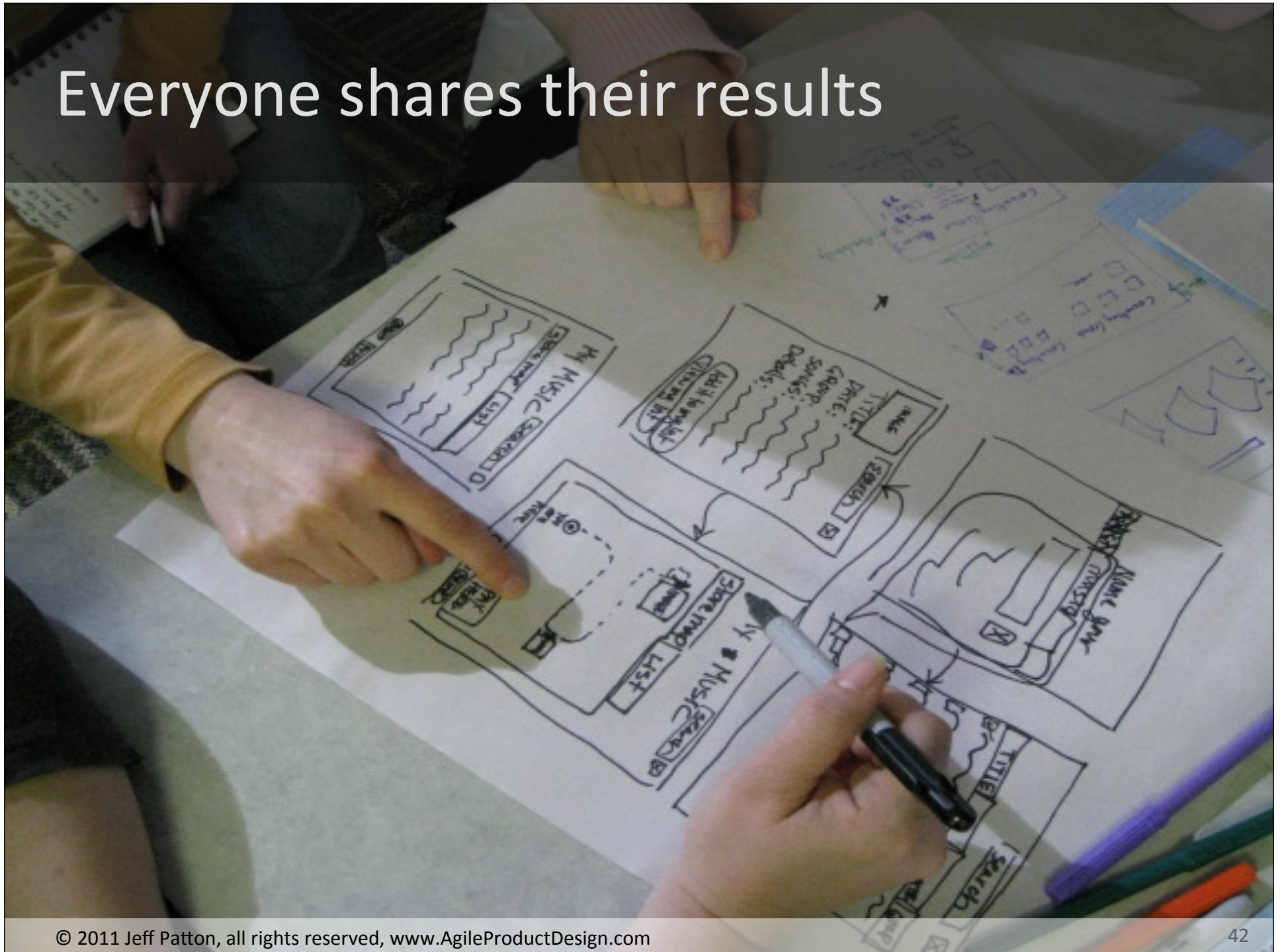
Sketch independently



Sketch independently



Everyone shares their results



Everyone shares their results



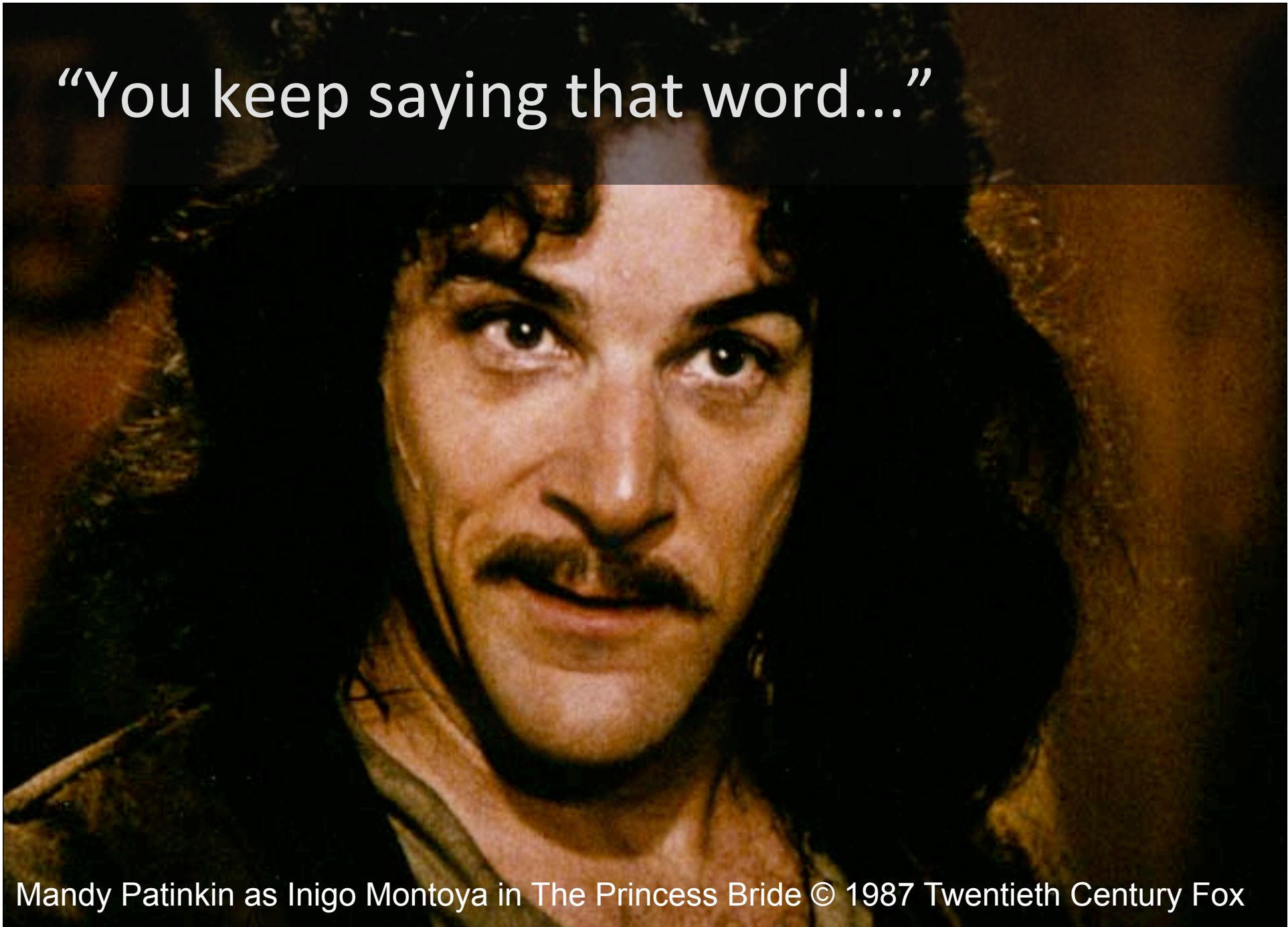
Find the best ideas (not the best artist)





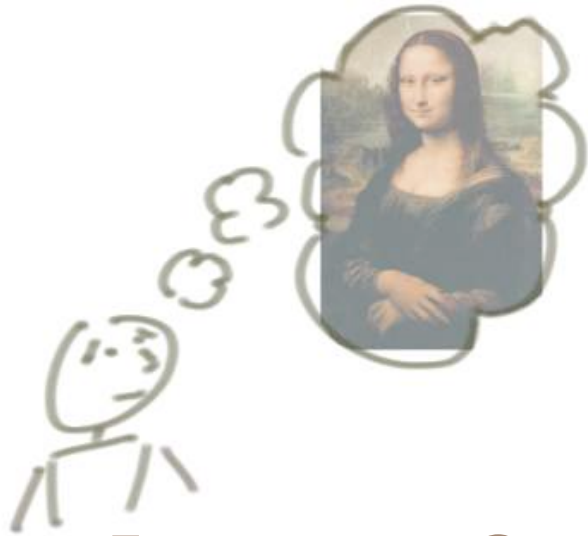
Iterate to validate and
refine solutions

“You keep saying that word...”



Mandy Patinkin as Inigo Montoya in The Princess Bride © 1987 Twentieth Century Fox

“incrementing” builds a bit at a time



Incrementing calls for a fully formed idea.

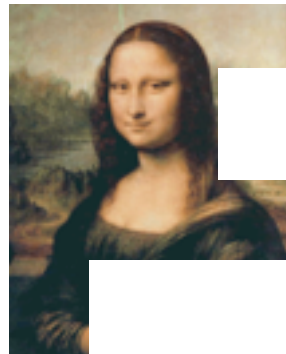
1



2



3



4



5



“iterating” builds a simple version,
validates it, then slowly builds up quality



A more iterative allows you to move from vague idea to realization making course corrections as you go.

1



2



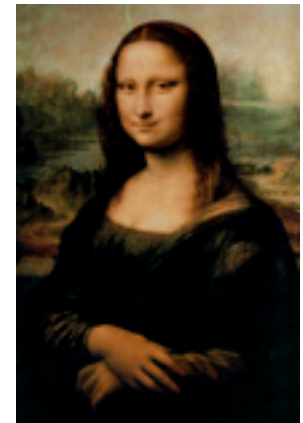
3



4



5

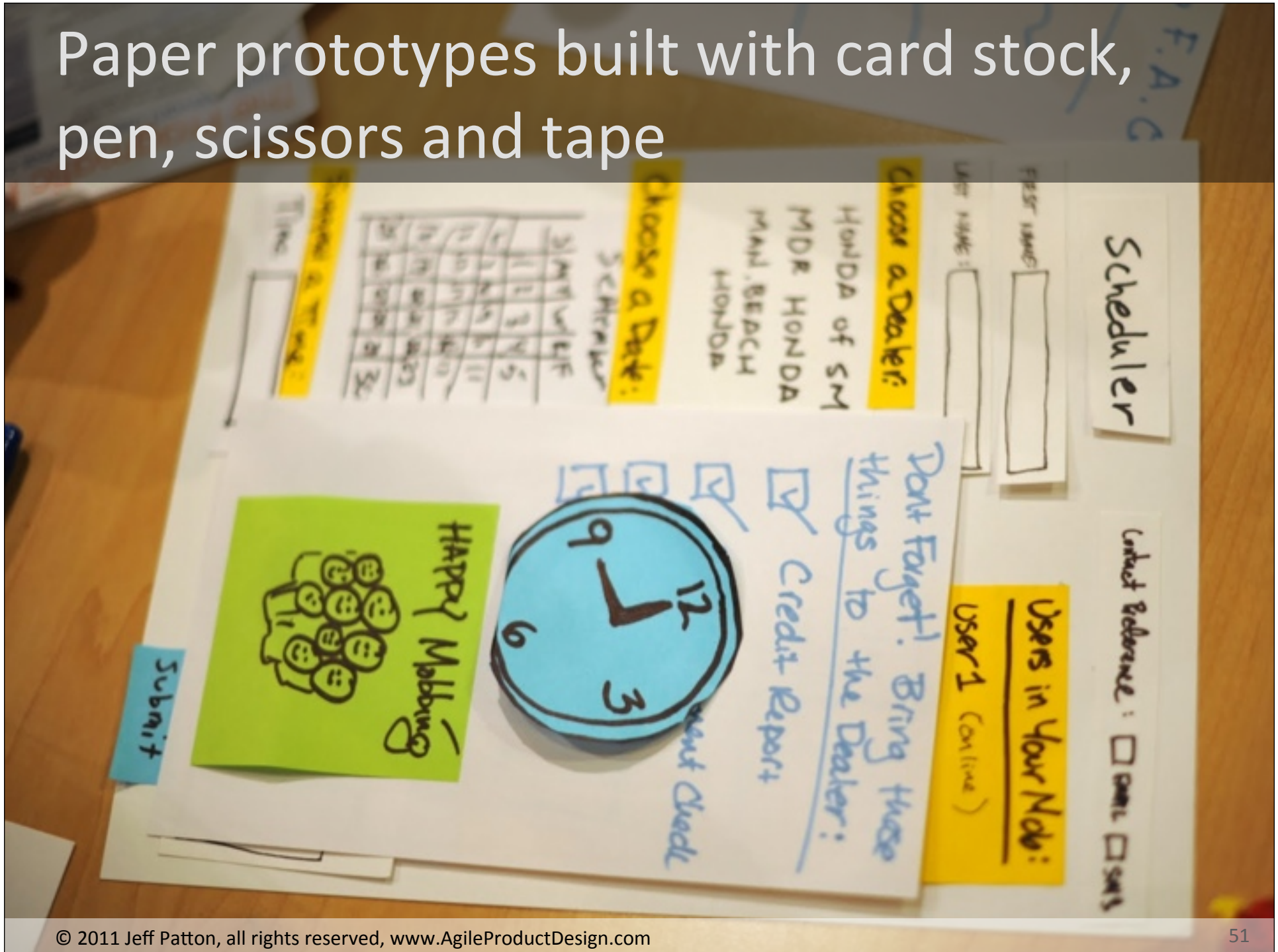


Validate the concept with a
lightweight prototype

Paper prototypes built with card stock, pen, scissors and tape



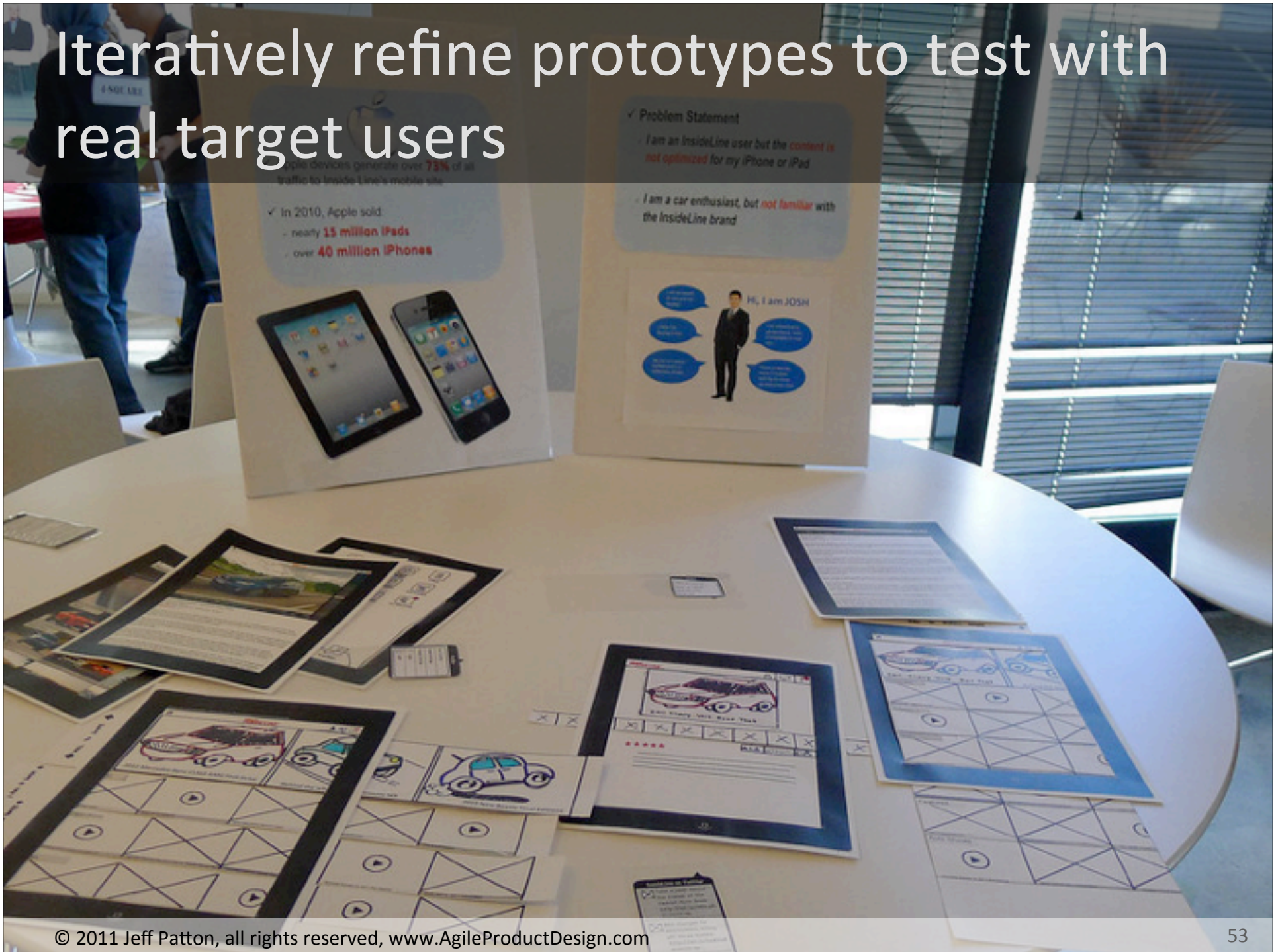
Paper prototypes built with card stock, pen, scissors and tape



“Testing” simulates use



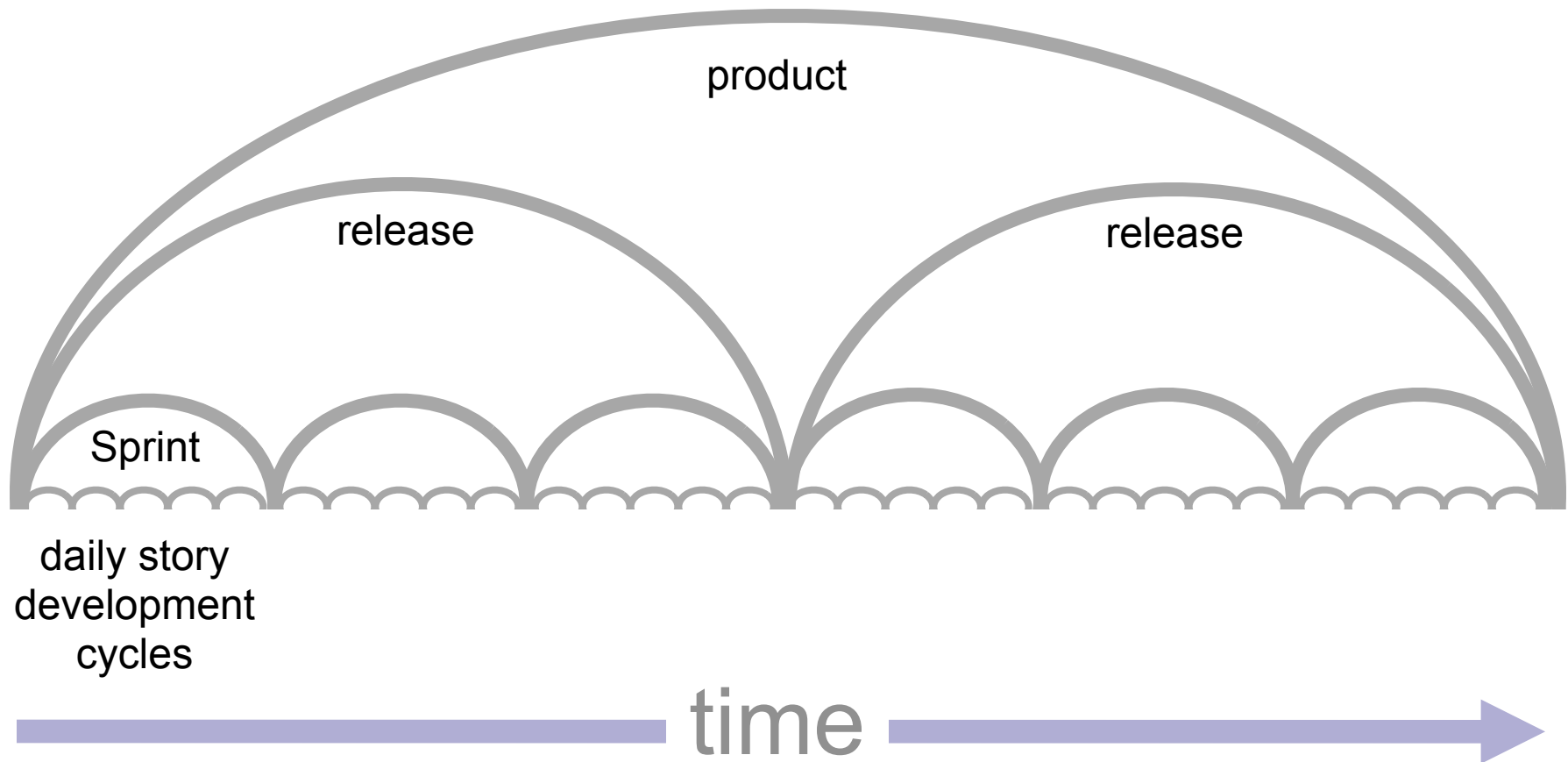
Iteratively refine prototypes to test with real target users



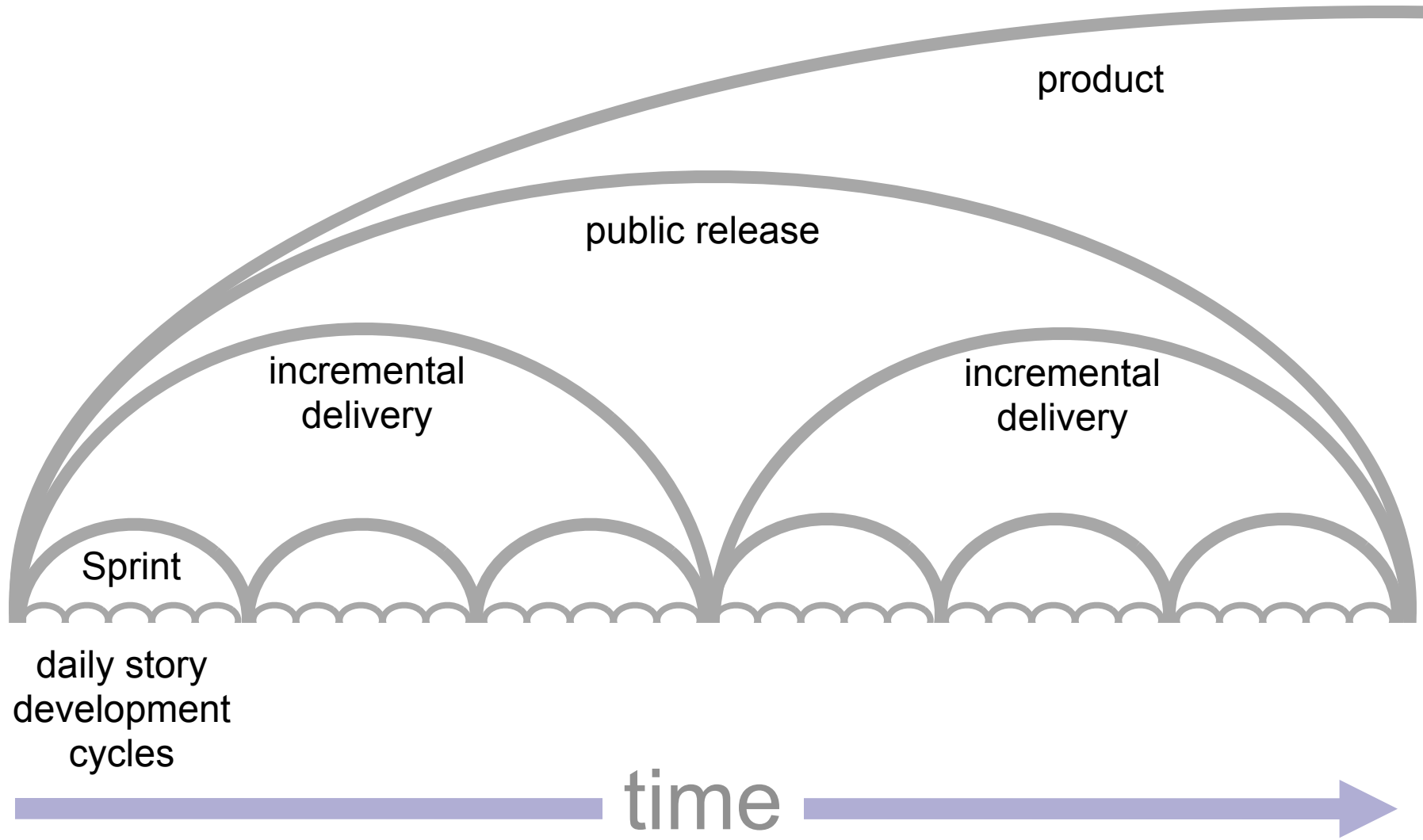
4

Plan to continue
learning as you deliver

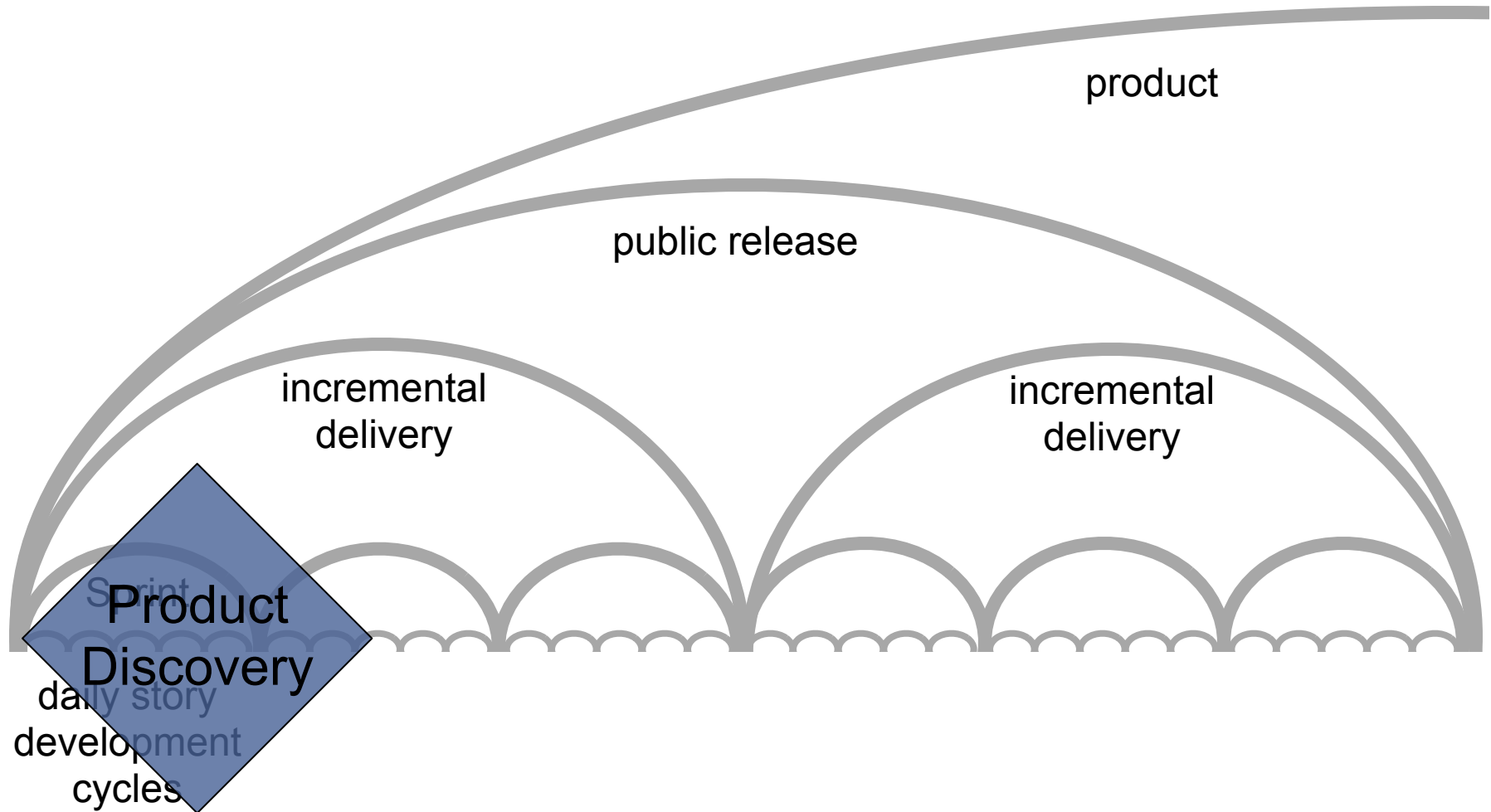
You'll need to plan beyond a single sprint



Your plan needs public and private releases - think “dress rehearsals”



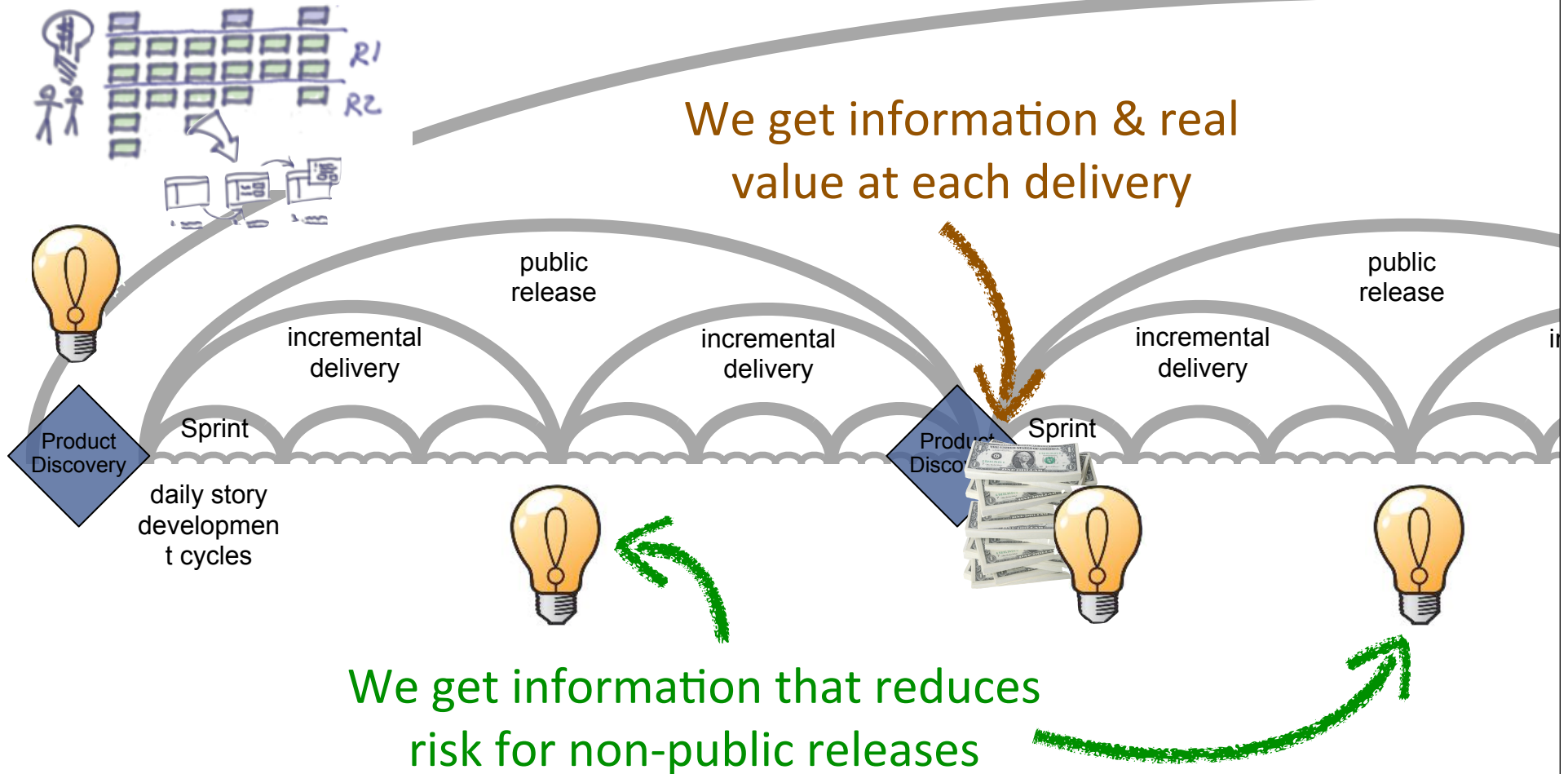
An initial discovery phase is where we first prototype and plan



Identify value to release and opportunities to learn

We get information & real value at each delivery

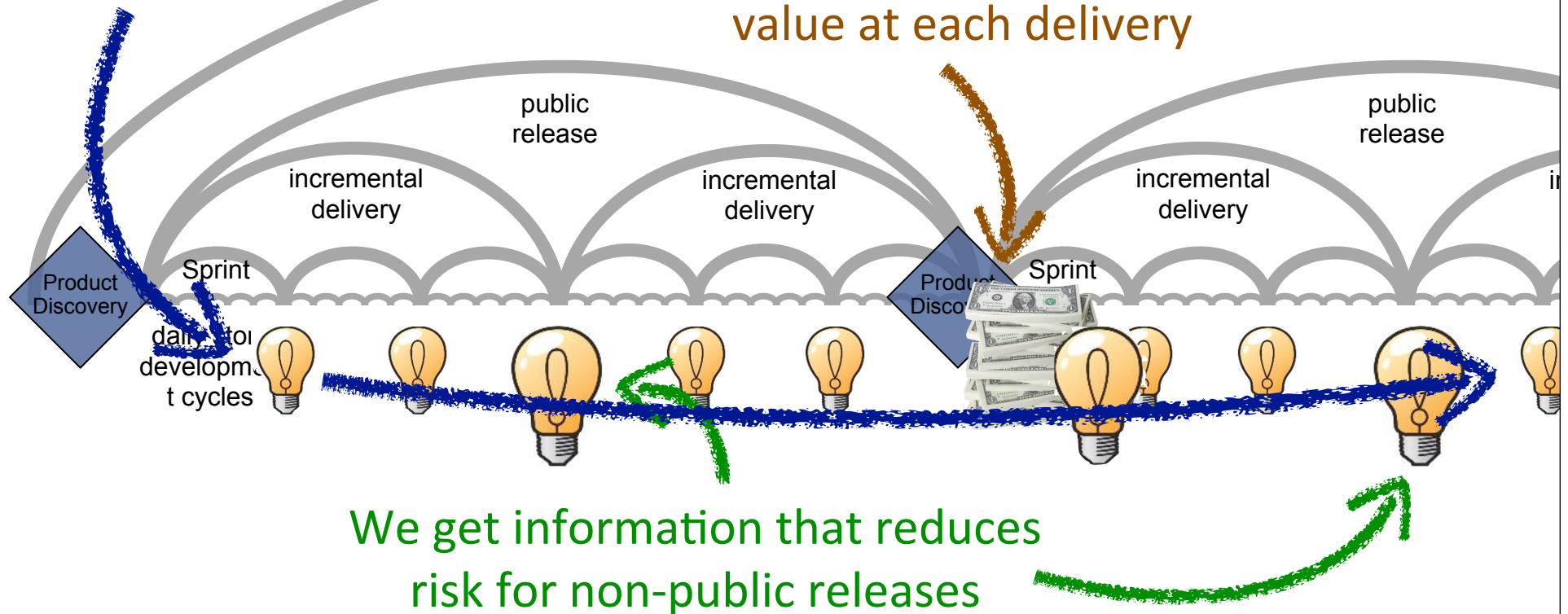
We get information that reduces risk for non-public releases



Use iterative and incremental development to learn and reduce risk

We learn a bit more and reduce risk with each sprint

We get information & real value at each delivery



We get information that reduces risk for non-public releases

Marshmallows



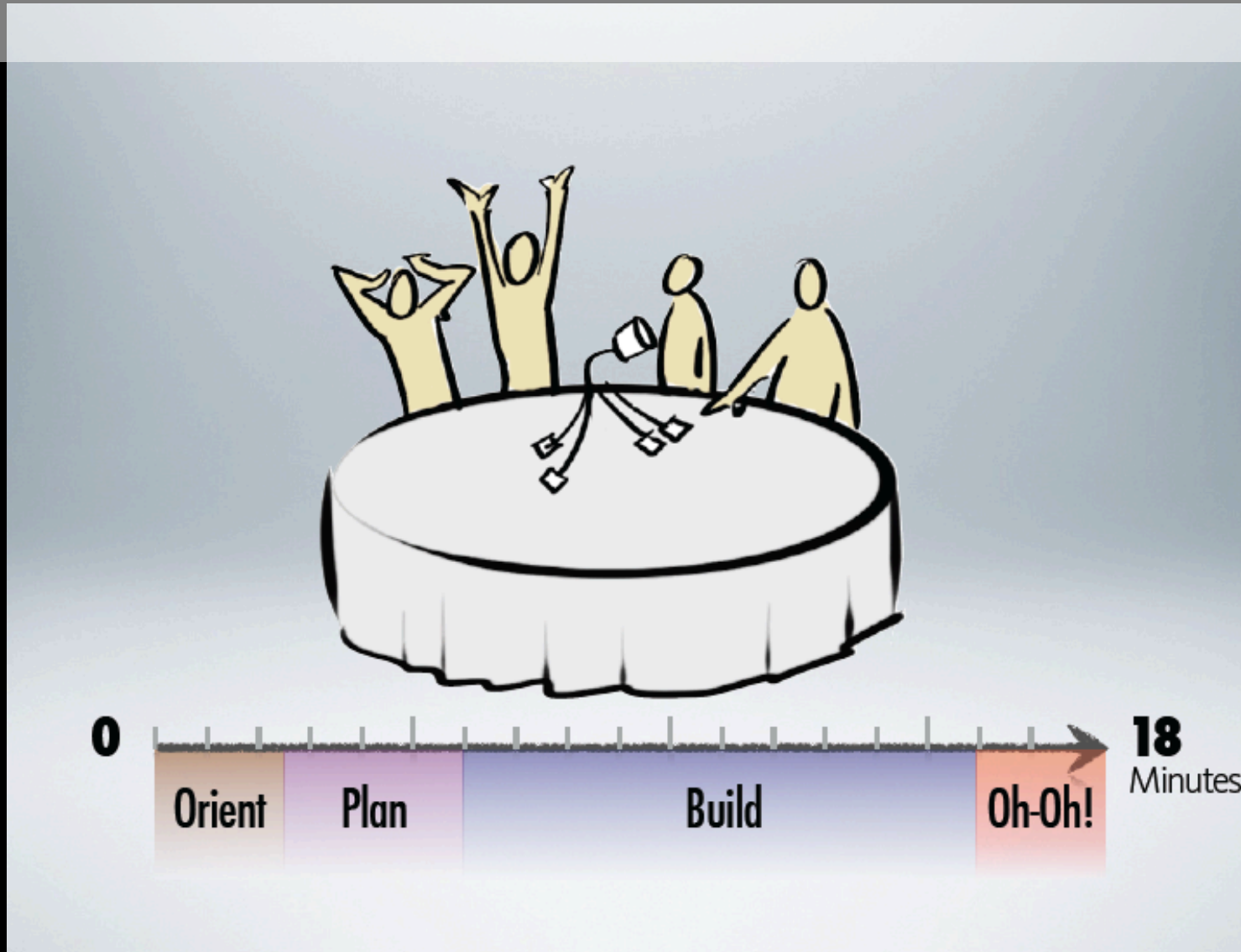
Marshmallows

The **Marshmallow Challenge**

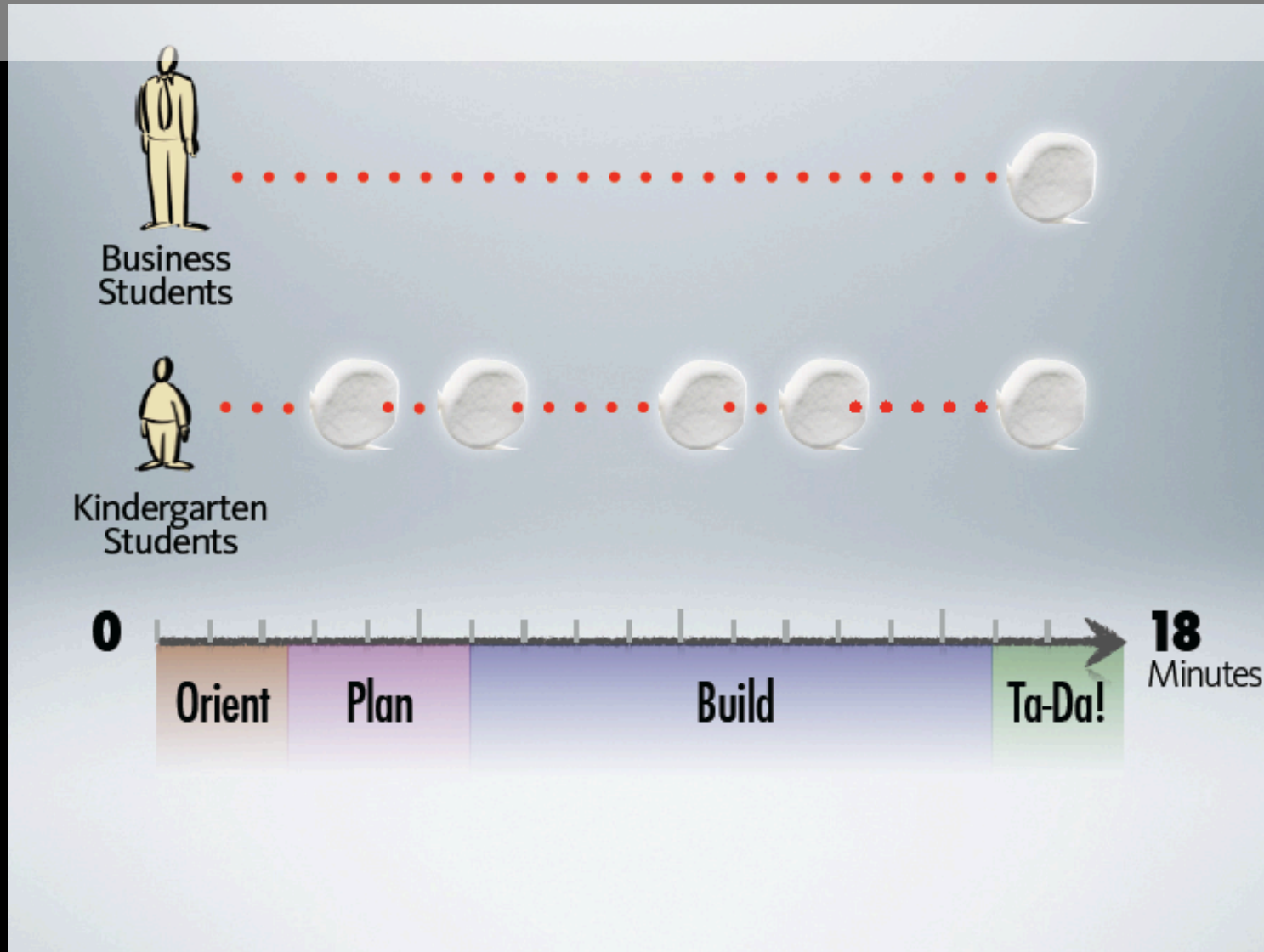
Tom Wujec
Autodesk



Marshmallows



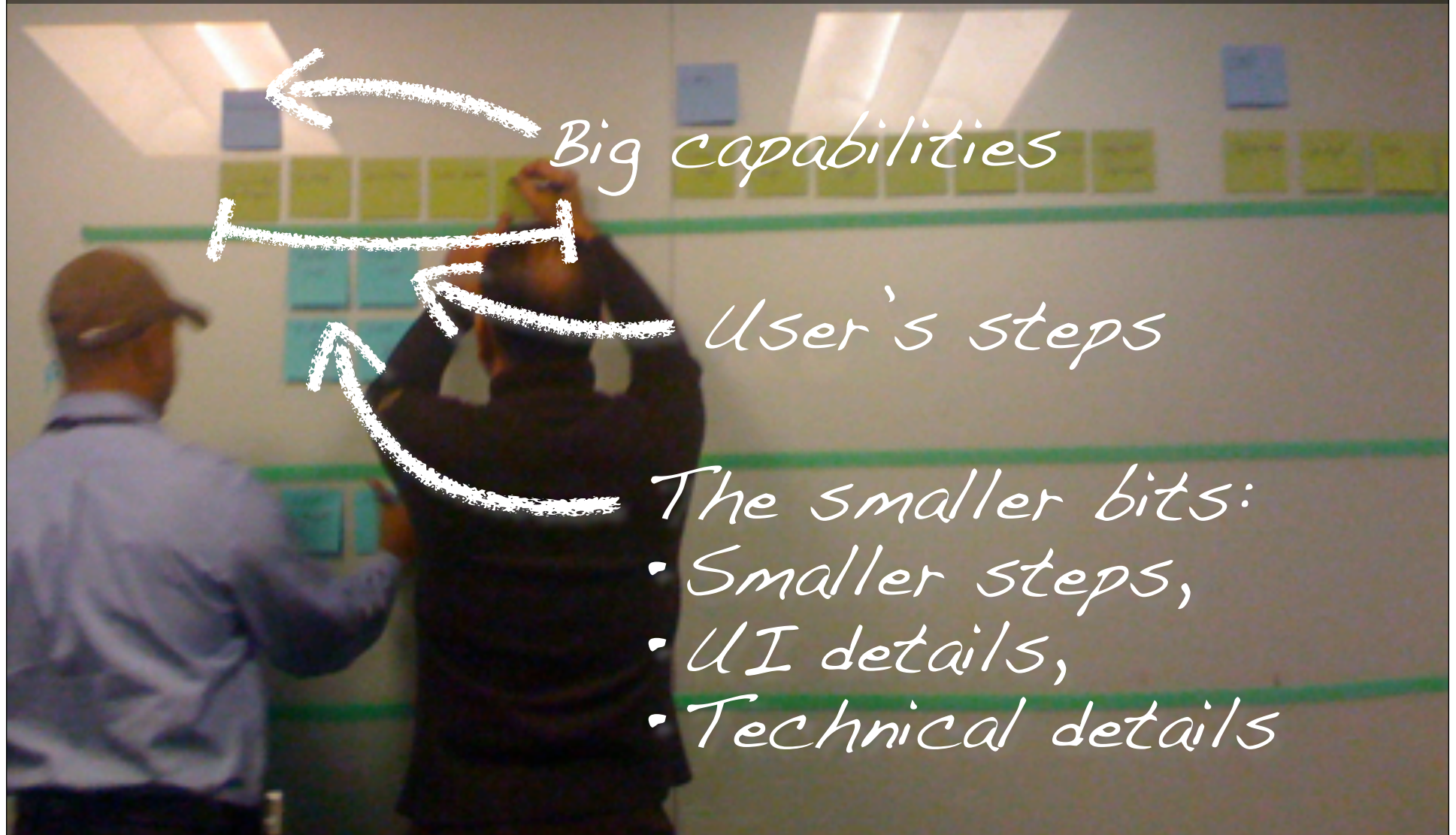
Marshmallows



Story maps tell the product's story,
including its incremental release plan



Story maps decompose the user's experience into product parts



Story Maps have latitude and longitude

the backbone

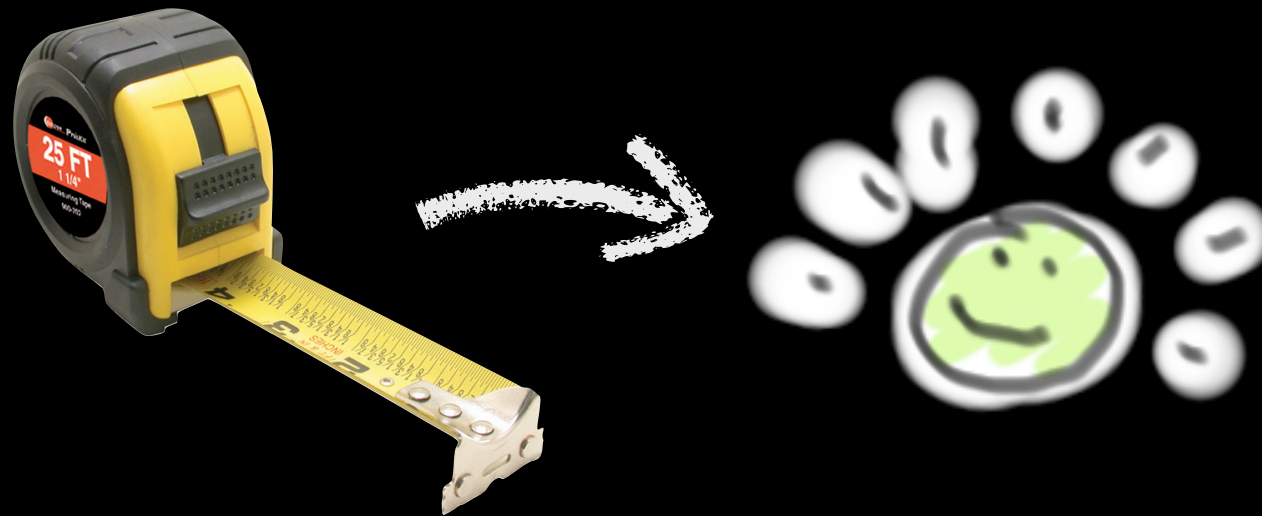
time
(from the user's perspective)

priority
*(time from the planner
and builder's perspective)*

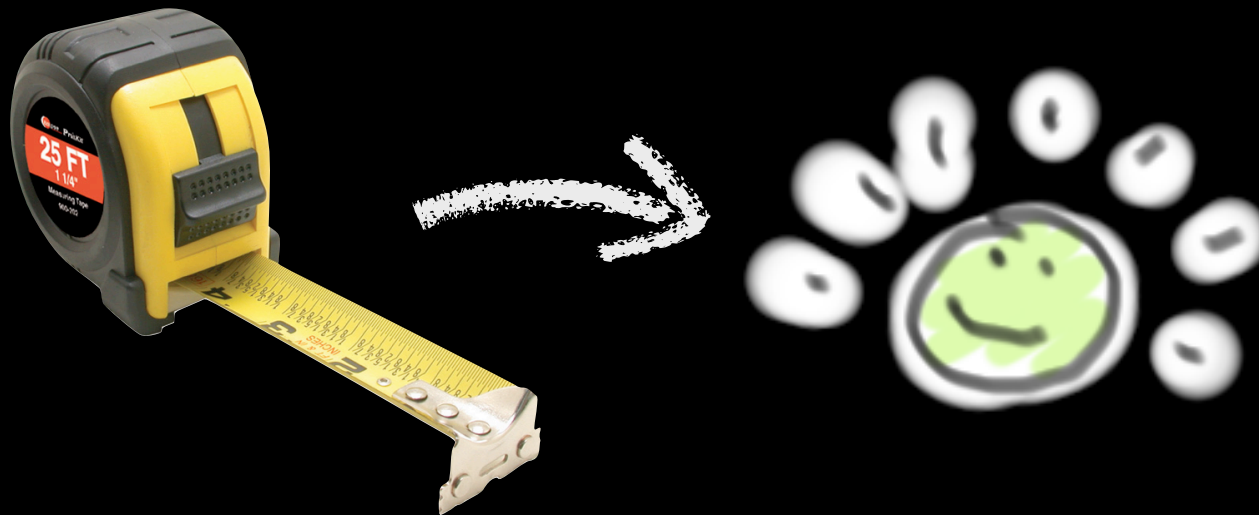
Plan by slicing the map into holistic valuable releases



If your product was
successful, what would you
measure?



In your product's road map,
pair **output** with **outcome**



Tell your product's story often



Telling your product story helps find problems



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



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



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



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



Today Edmunds' process includes routine all-hands collaborative discovery work





Leah Buley

<http://www.adaptivepath.com/aboutus/leah.php>

“Design isn’t something that designers produce, design is a process that designers facilitate.”

--Leah Buley, Adaptive Path



Leisa Reichelt

www.disambiguity.com

“Design by community is not design by committee”

--Leisa Reichelt

Using a collaborative design thinking approach goes best when you...

Incorporate effective facilitation

Get support and facilitation from someone who understands design practice (at least initially)

Aggressively time-box

Keep work and progress visible

Encourage a culture of open communication and shared ownership



Value only comes after
delivery

Optimize the system to
shorten the time from
opportunity identification
to **observable outcome**



Using Design Thinking

to

Questions?

Stop Building Worthless Software

Thanks to Edmunds.com for agreeing to share photos and their story.

download slides at:

http://www.agileproductdesign.com/downloads/patton_design_thinking_slides.pdf



Jeff Patton

jpatton@acm.org

twitter: @jeffpatton

www.agileproductdesign.com

Further reading:

Read Jeff Patton's article: Design Thinking: 4 Steps To Better Software: http://www.stickyminds.com/s.asp?F=S16503_COL_2

Watch the IDEO Shopping Cart Video that motivates many people to try a Design Thinking Approach: <http://vimeo.com/16456835>

Watch Frog Design's Robert Fabricant speak about output vs. outcome: <http://www.ixda.org/resources/robert-fabricant-behavior-our-medium>

Tim Brown's Lecture: Innovation Through Design Thinking: <http://mitworld.mit.edu/video/357/>

Beyond Sprint Zero presentation slides describe product discovery: http://www.agileproductdesign.com/downloads/beyond_sprint_zero.pdf

Useful references:

User Research Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_user_investigtion.pdf

Pragmatic Persona Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_pragmatic_persona.pdf

User Story Mapping Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_story_mapping.pdf

Design Thinking Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_design_thinking.pdf

Design Studio Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_design_studio.pdf

Paper Prototyping Quickref: http://www.agileproductdesign.com/downloads/quickrefs/patton_paper_prototyping.pdf