

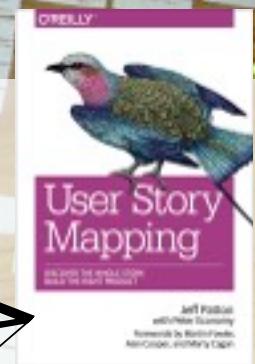
Story Mapping

discover the whole story



Jeff Patton
jeff@jpattonassociates.com
twitter: @jeffpatton

I wrote this
book!



Stupid stuff I used to believe about Agile stories:

1. Stories are way to document requirements in Agile processes
2. Good stories are small
3. Good product backlogs are prioritized lists of stories
4. Each story we build is valuable to customers and users



Documents don't
work the way you
think they do



Imagine a simple phone conversation...



Jen Yates' Cake Wrecks: www.cakewrecks.com

Specifying in writing doesn't work well

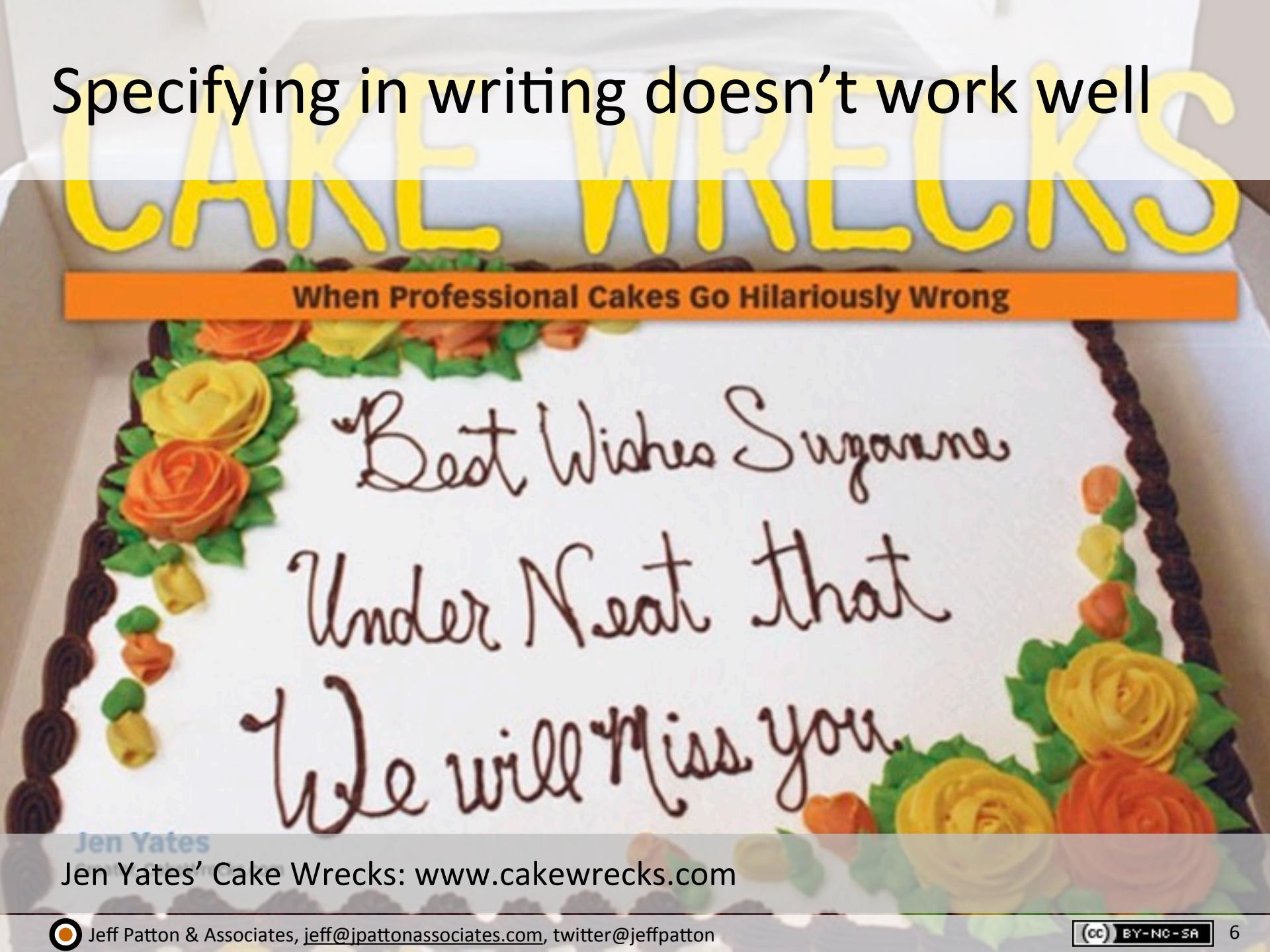


The screenshot shows the homepage of the Cake Wrecks website. The header features a large, stylized title "CAKE WRECKS" with a subtitle "WHEN PROFESSIONAL CAKES GO HORRIBLY, HILARIOUSLY WRONG". Below the header is a navigation bar with links to Home, FAQ, Press, Contact, About, and Stuff. To the right of the navigation are social media icons for RSS, Facebook, and Twitter. The main content area includes a "What The H?" section from October 16, 2012, with a photo of a cake decorated with green frosting and small figures. A sidebar on the left lists various blog posts from the BlogHer network, including Sun-Dried Tomato Pesto, Grilled Chicken Pitas, Cucumber Yogurt Dressing, Kala Channa Subzi, and Motherhood Before and After Delivery. On the right side of the page is a photograph of Jen Yates' book "CAKE WRECKS: When Professional Cakes Go Hilariously Wrong". The book cover features the same title in large yellow letters and a photo of a cake with a message that reads: "Best Wishes Suzanne Under Neat that We will Miss you". The author's name, Jen Yates, and her website, cakewrecks.com, are also visible on the book cover.

<http://www.cakewrecks.com/>

Cake Wrecks, book by Jen Yates,

Specifying in writing doesn't work well



Jen Yates

Jen Yates' Cake Wrecks: www.cakewrecks.com



Jeff Patton & Associates, jeff@jpattonassociates.com, [@jeffpatton](https://twitter.com/jeffpatton)

(cc) BY-NC-SA

Specifying in writing doesn't work well



Jen Yates' Cake Wrecks: www.cakewrecks.com



Jeff Patton & Associates, jeff@jpattonassociates.com, [twitter@jeffpatton](https://twitter.com/jeffpatton)



Specifying in writing doesn't work well



Jen Yates' Cake Wrecks: www.cakewrecks.com



Jeff Patton & Associates, jeff@jpattonassociates.com, [twitter@jeffpatton](https://twitter.com/jeffpatton)



Specifying in writing doesn't work well



Jen Yates' Cake Wrecks: www.cakewrecks.com

Specifying in writing doesn't work well



Jen Yates' Cake Wrecks: www.cakewrecks.com



Jeff Patton & Associates, jeff@jpattonassociates.com, [twitter@jeffpatton](https://twitter.com/jeffpatton)



Specifying in writing doesn't work well

The Washington Post | Politics | Opinions | Local | Sports | National | World | Business | Tech | Lifestyle | Entertainment | Jobs | More

HOME INDEX SEARCH ARCHIVES washingtonpost.com NEWS STYLE SPORTS CLASSIFIEDS MARKETPLACE

Space Exploration SPECIAL REPORT

PRINT EDITION TOP NEWS WORLD NATION POLITICS METRO BUSINESS & TECH

Space Space Report

Partner Sites: Newsweek.com Britannica Internet Guide

Mystery of Orbiter Crash Solved

By Kathy Sawyer
Washington Post Staff Writer
Friday, October 1, 1999; Page A1

NASA's Mars Climate Orbiter was lost in space last week because engineers failed to make a simple conversion from English units to metric, an embarrassing lapse that sent the \$125 million craft fatally close to the Martian surface, investigators said yesterday.

Officials are scrambling to determine whether a similar error is buried in the computer files of two other spacecraft currently cruising through space: the Mars Polar Lander, scheduled to hit the Martian surface on Sunday, and the Deep Impact comet mission.

“... engineers failed to make a simple conversion between English units and metric, an embarrassing lapse...”



Scientists do not yet know what caused the Mars Orbiter to crash. (AP)

Sometimes mistakes are less funny

When we share and sign off a document we may believe we understand



I'm glad we all agree.

Kent has a disruptively simple idea



Stop it.

Stop exchanging documents.

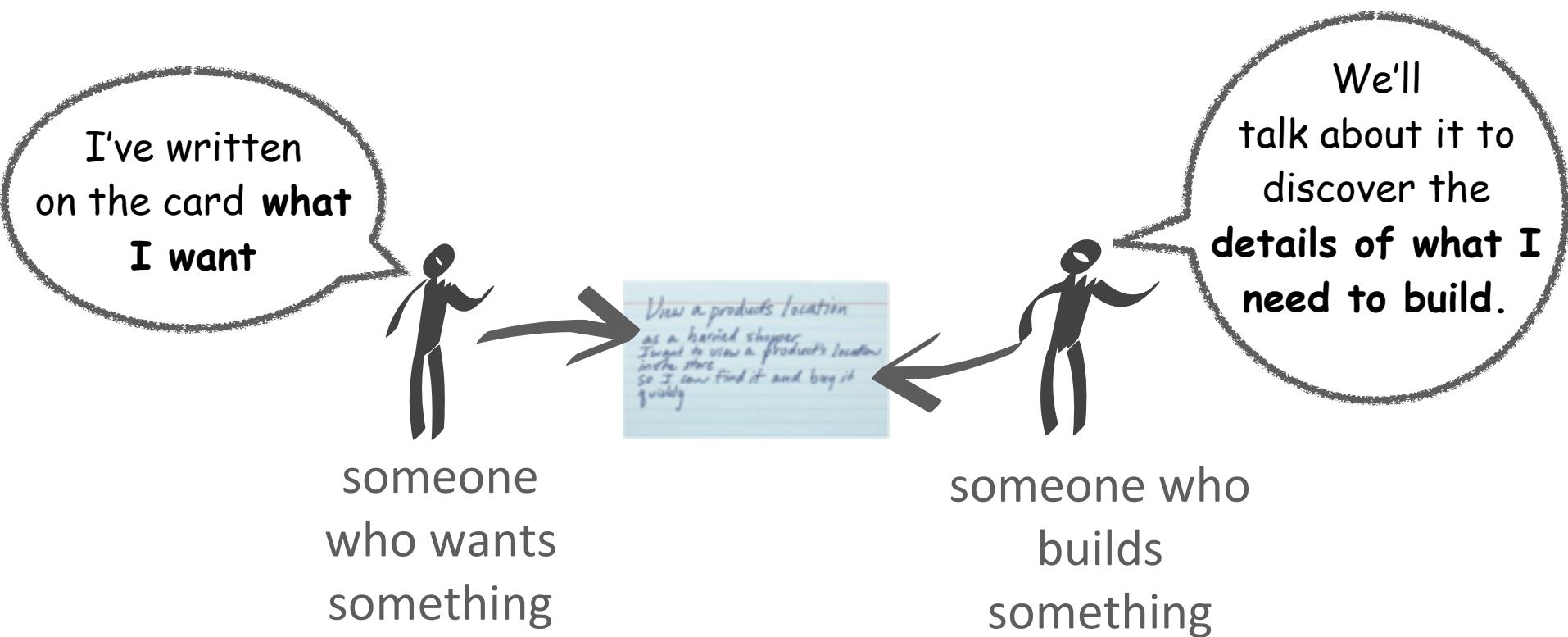
Tell me your story.



If we we could
just talk about this, we
could figure it out
together.



The original idea of a story was simple: use it to facilitate a conversation



Stories get their name
from how we use
them, not how we
write them.

But, we still managed
to screw that up



This is a Scrum backlog grooming session

blah blah
blahdy-blah
b'blah blah
blahdy-blah blah
blah blahdy-blah
b'blah blah
blahdy-blah blah
blah blahdy-blah
b'blah blah
blahdy-blah blah
blah blah
blahdy-blah blah

This is JIRA
projected on
the wall

About three of
10 people
actively engage

He's not raising
his hand to speak,
he's yawning

He's secretly
reading email on
his smart phone



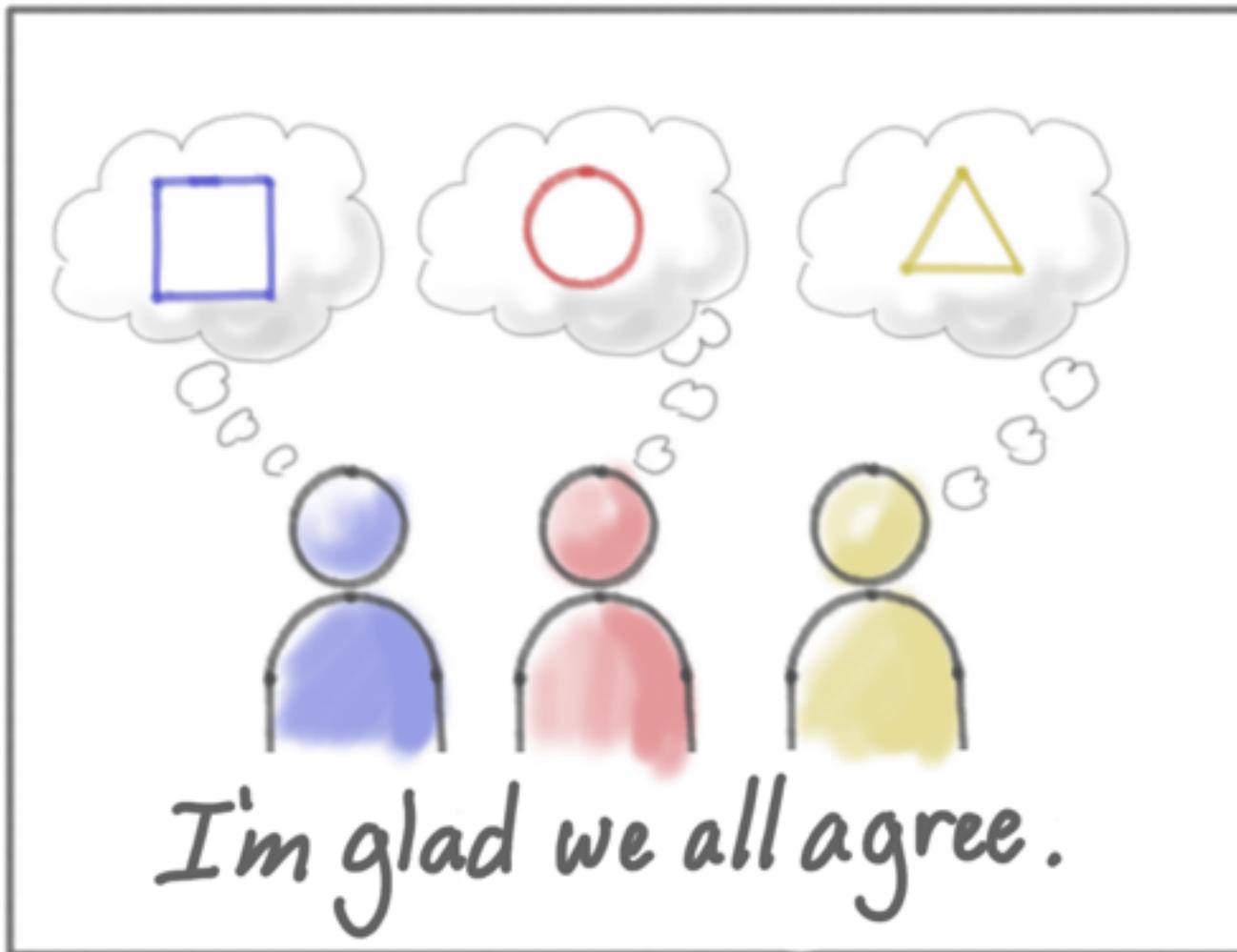
This isn't the kind of
conversation Kent
had in mind



Something special is
going on during an
effective conversation



With a shallow discussion, we may all take away something different



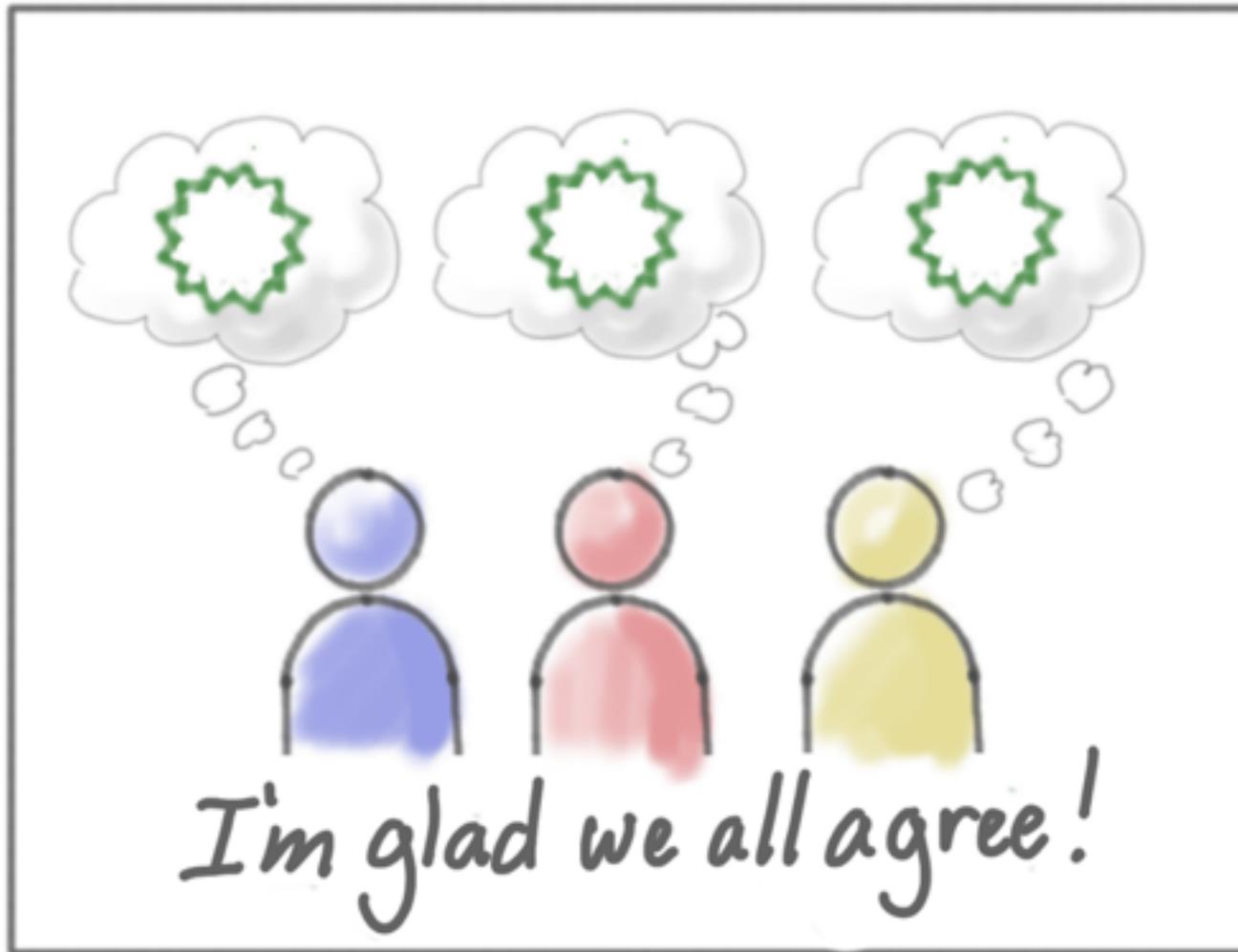
When we externalize our thinking with words and pictures, we detect differences



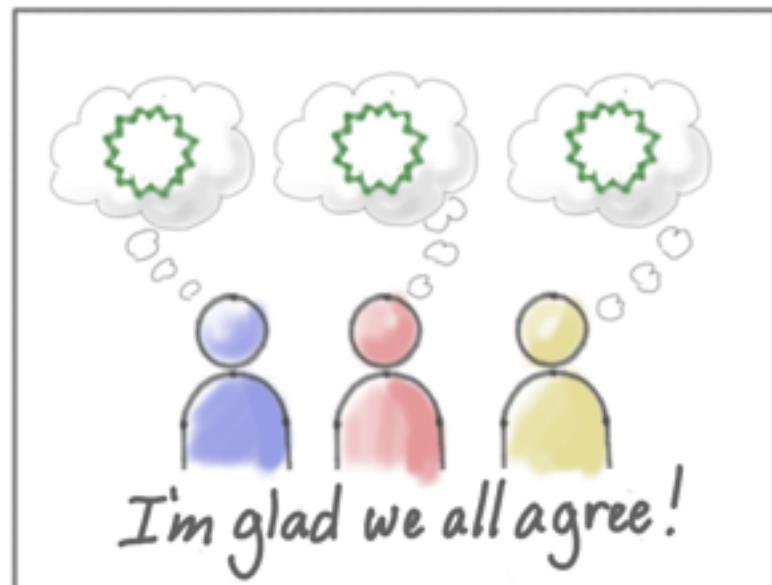
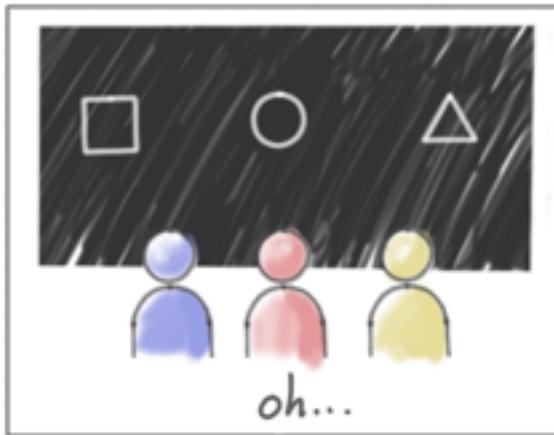
When we combine and refine, we arrive at something better



Afterwards, when we say the same thing, we actually mean it



Shared understanding and alignment are the objectives of collaborative work



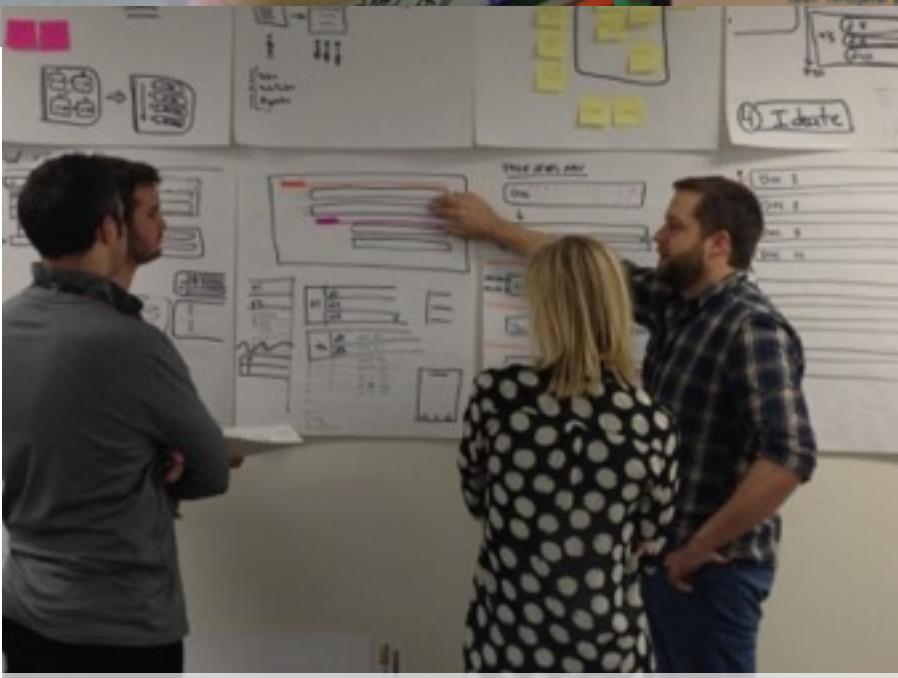
* Credit for this illustration goes to ThoughtWorks' Luke Barret. Jeff Patton drew these illustrations based on Luke's. Luke doesn't recall where he first saw this cartoon.



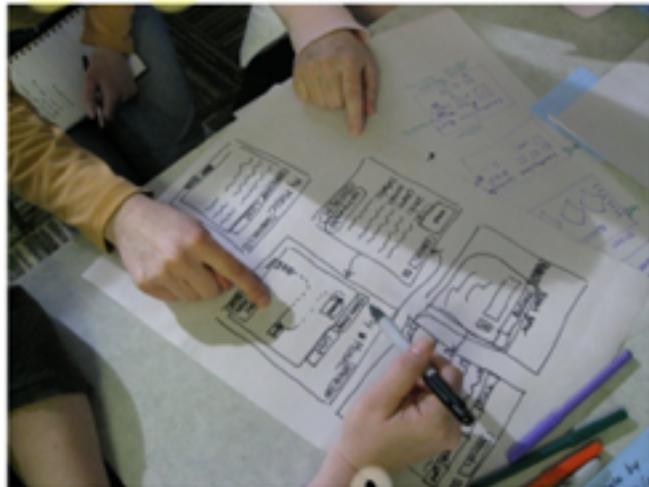
Words and pictures help everyone build shared understanding



To build shared understanding, use sketching and recording on walls and whiteboards

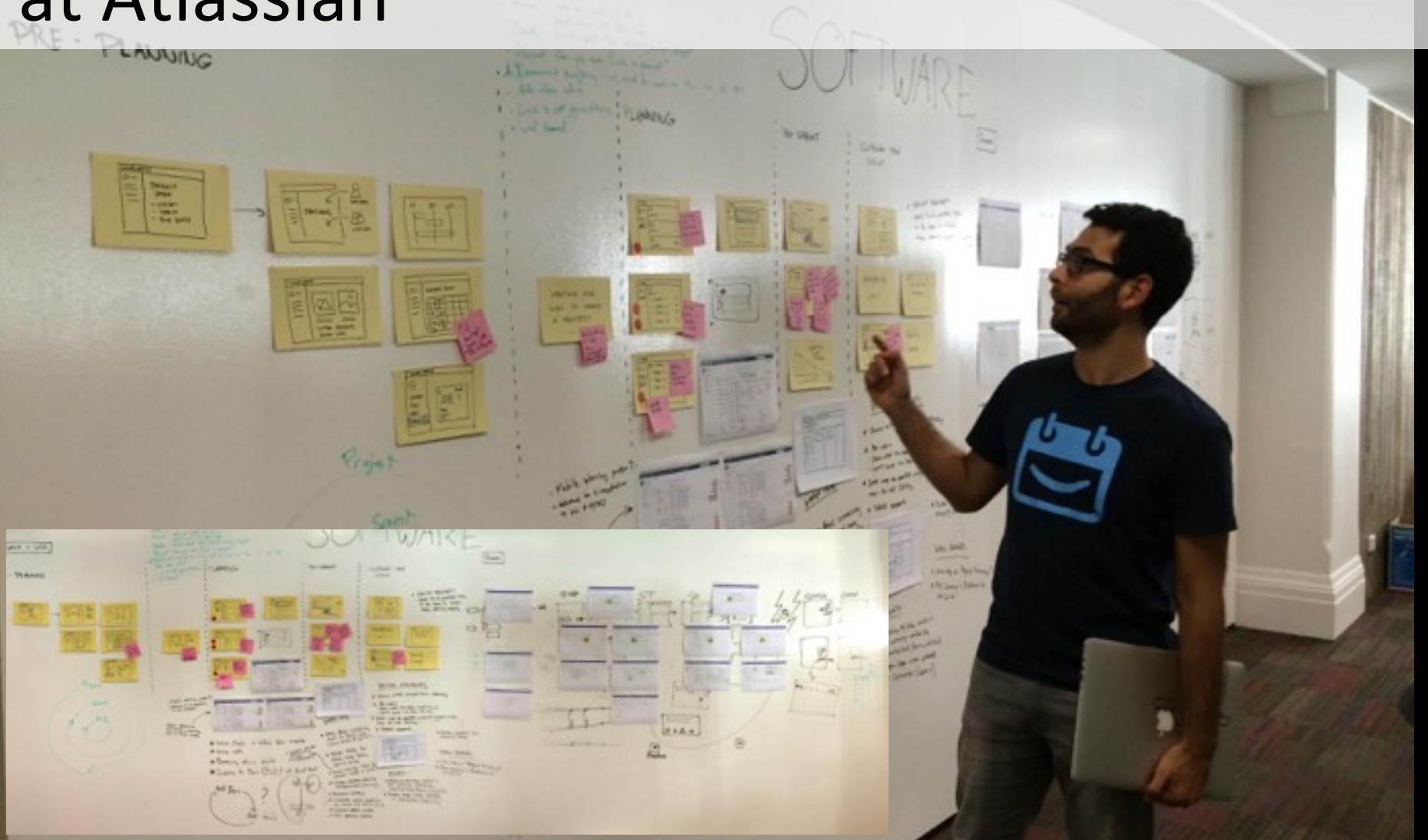


Use words and pictures



sketch,
tell stories,
write down
facts & decisions

Shared Understanding and collaboration at Atlassian



Shared Understanding and collaboration at Atlassian



Shared Understanding and collaboration at Atlassian



What you record during conversations works like a vacation photo



Looking at it helps you remember details that aren't in the photo



What you record during conversations works like a vacation photo



Looking at it helps you remember details that aren't in the photo



Effective story conversations **build shared understanding**

The best **documents** use words
and pictures to **help recall our**
conversations, they don't replace
conversations



You'll have to think
things through

This is a cake for a baby shower



Jen Yates' Cake Wrecks: www.cakewrecks.com

I don't think they
thought this through...

This is a cake for a baby shower

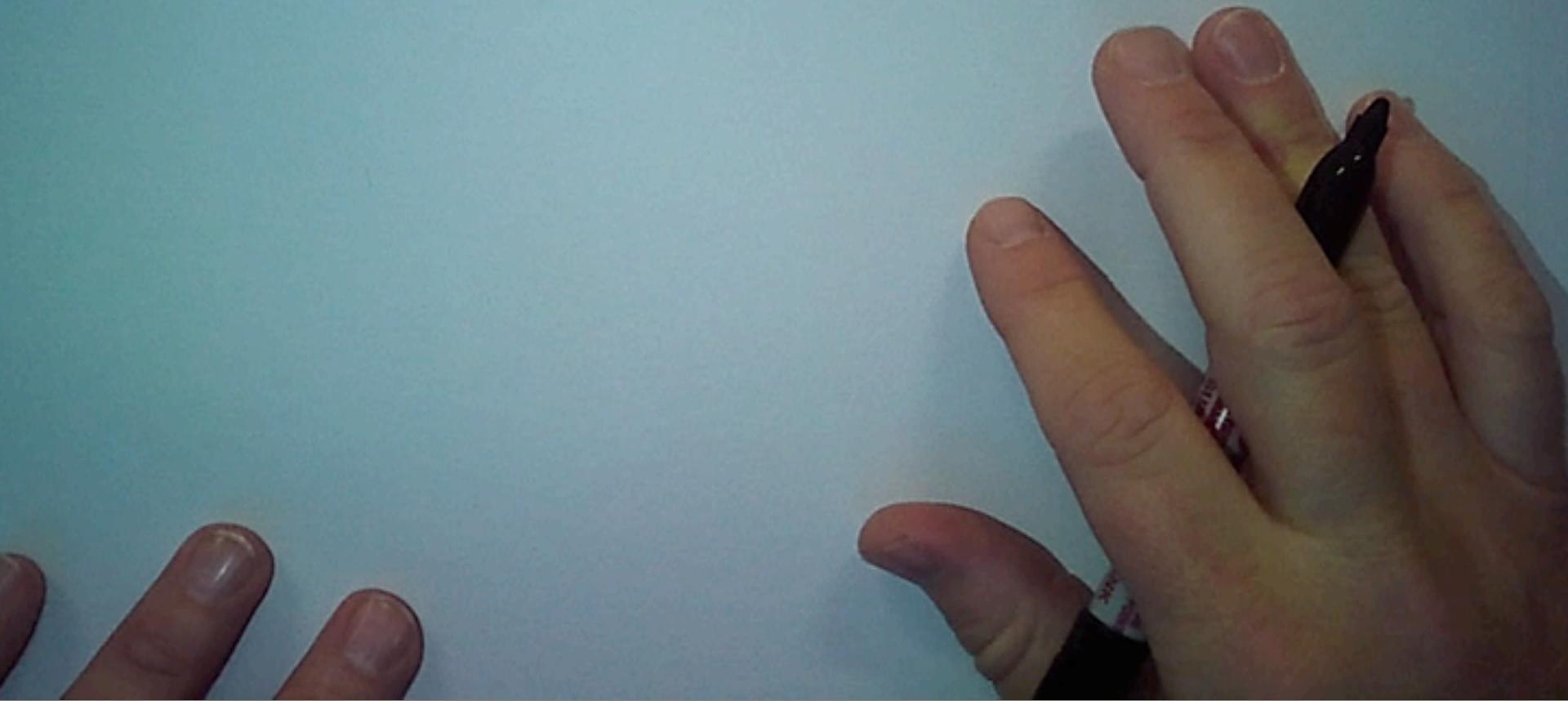


Jen Yates' Cake Wrecks: www.cakewrecks.com



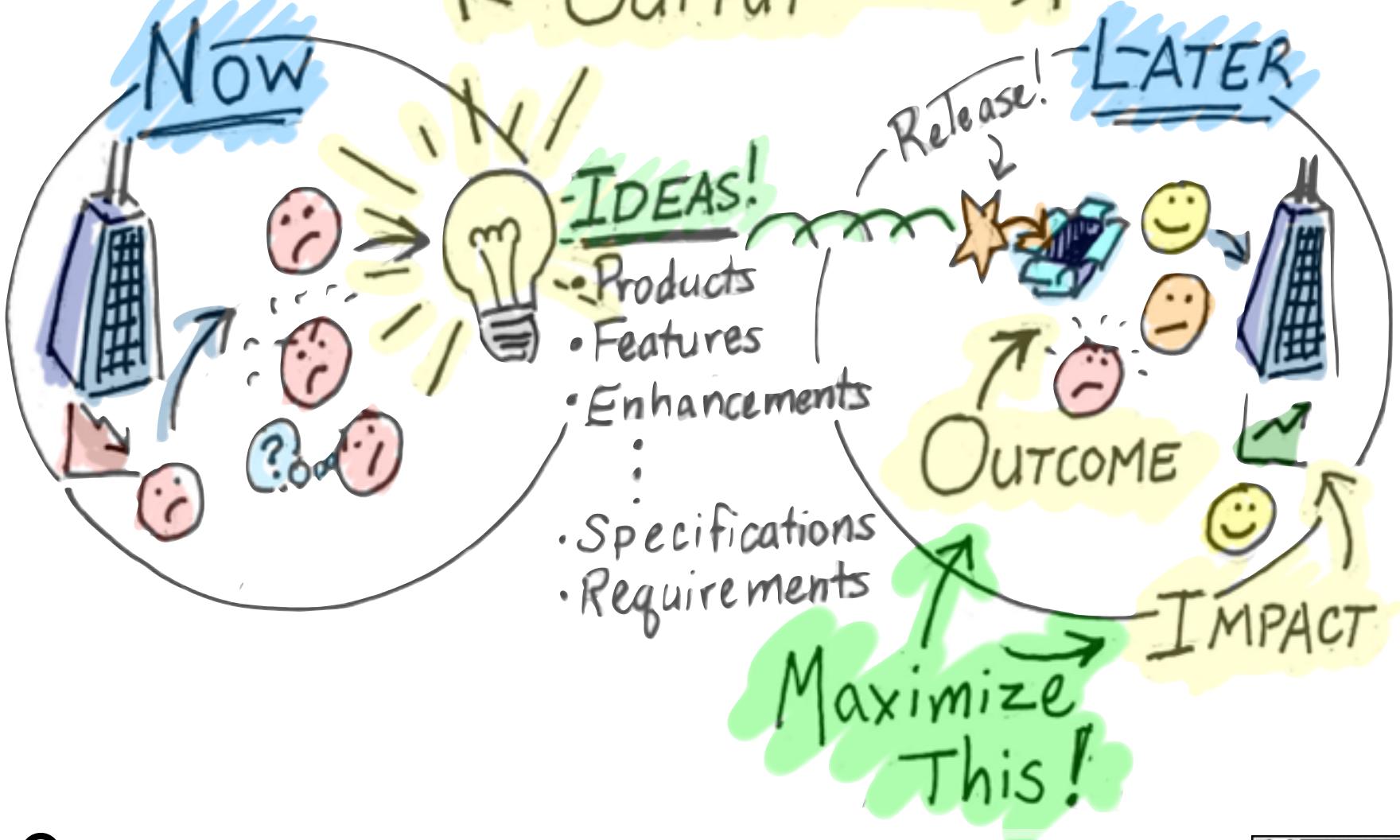
Jeff Patton & Associates, jeff@jpattonassociates.com, [twitter@jeffpatton](https://twitter.com/jeffpatton)





Minimize This →

← OUTPUT →



Talk about the outcome, not just the output

output



we build this

outcome



we want this

Stories are an antidote to “requirements”



Stories

Plan using units of change, not requirements. For example, if you want to move traffic from one part of your system to another, plan the traffic with the same responsibility as moving users to dial frequently used numbers. This will help you to estimate the development effort necessary to implement it.

Software development has been steered wrong by the word “requirement”, defined in the dictionary as “something mandatory or obligatory.” The word carries a connotation of absolutism and permanence, inhibitors to embracing change. And the word “requirement” is just plain wrong. Out of one thousand pages of “requirements”, if you deploy a system with the right 20% or 10% or even 5%, you will likely miss all of the business benefit envisioned for the whole system. So what’s the problem? Requirements are “mandatory or obligatory.” They’re not stories. Stories are “optional.”



Kent suggested we
talk about what
happens when things
come out

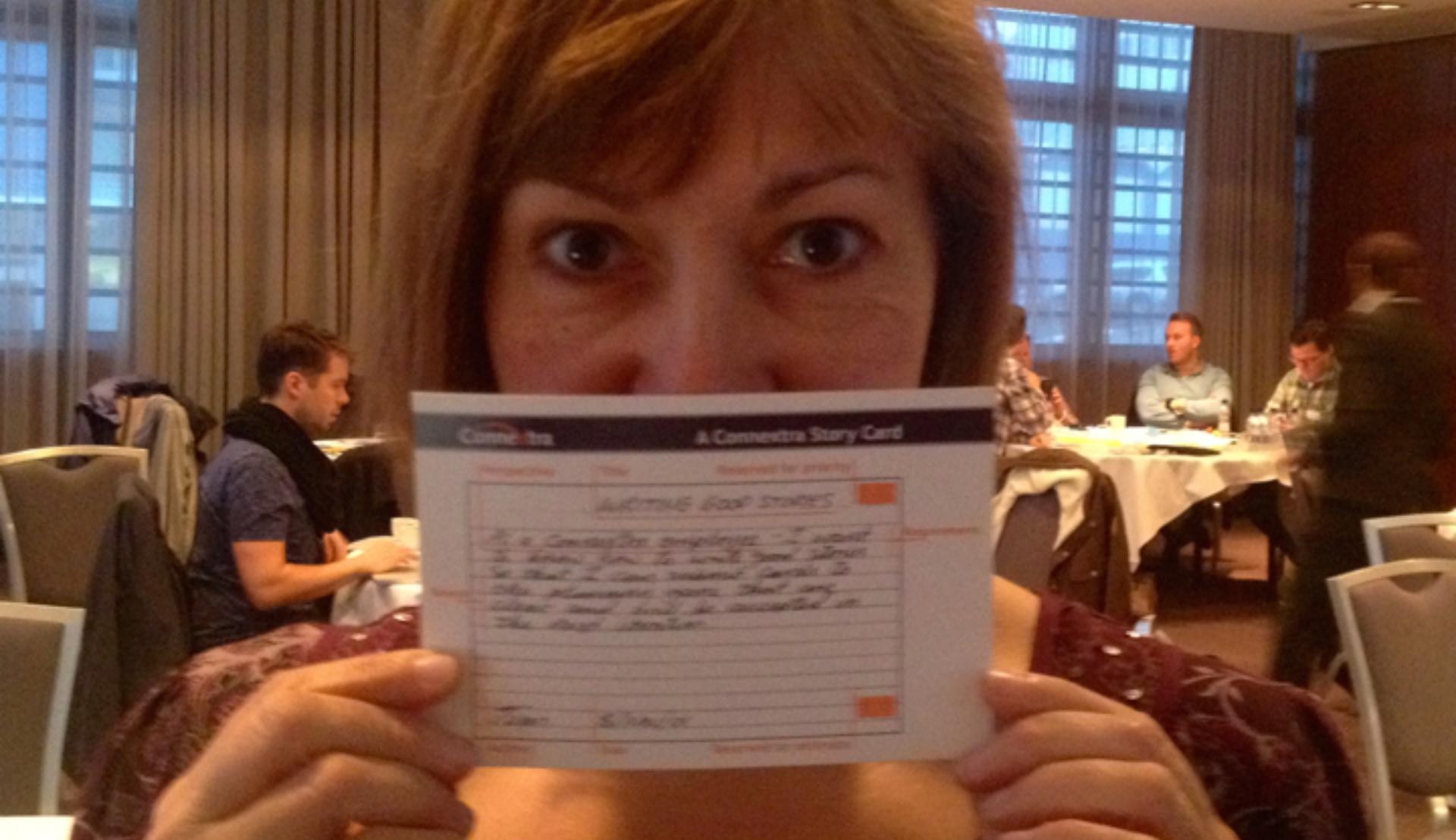
Talk about who does what, and why

What I was thinking
of was the way users sometimes
tell stories about the cool new things the
software they use does:

*“I type in the zip code and it
automatically fills in the city and state
without me having to touch a button!”*

I think that was the example that triggered the idea.
If you can tell stories about what the software does
and **generate energy and interest and a vision in
your listener's mind**, then why not tell
stories before the software does it?

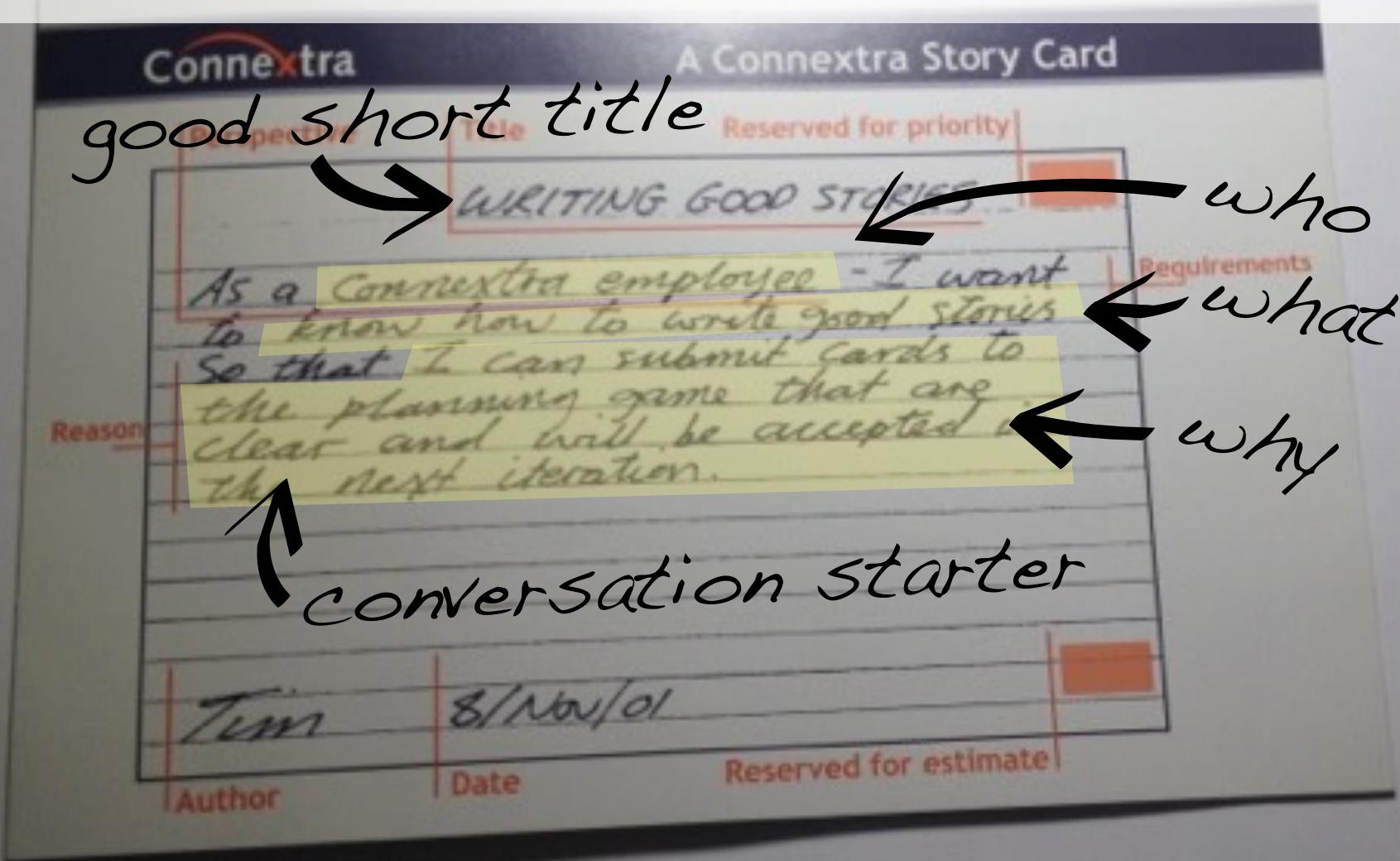




Say “Hi” to Rachel

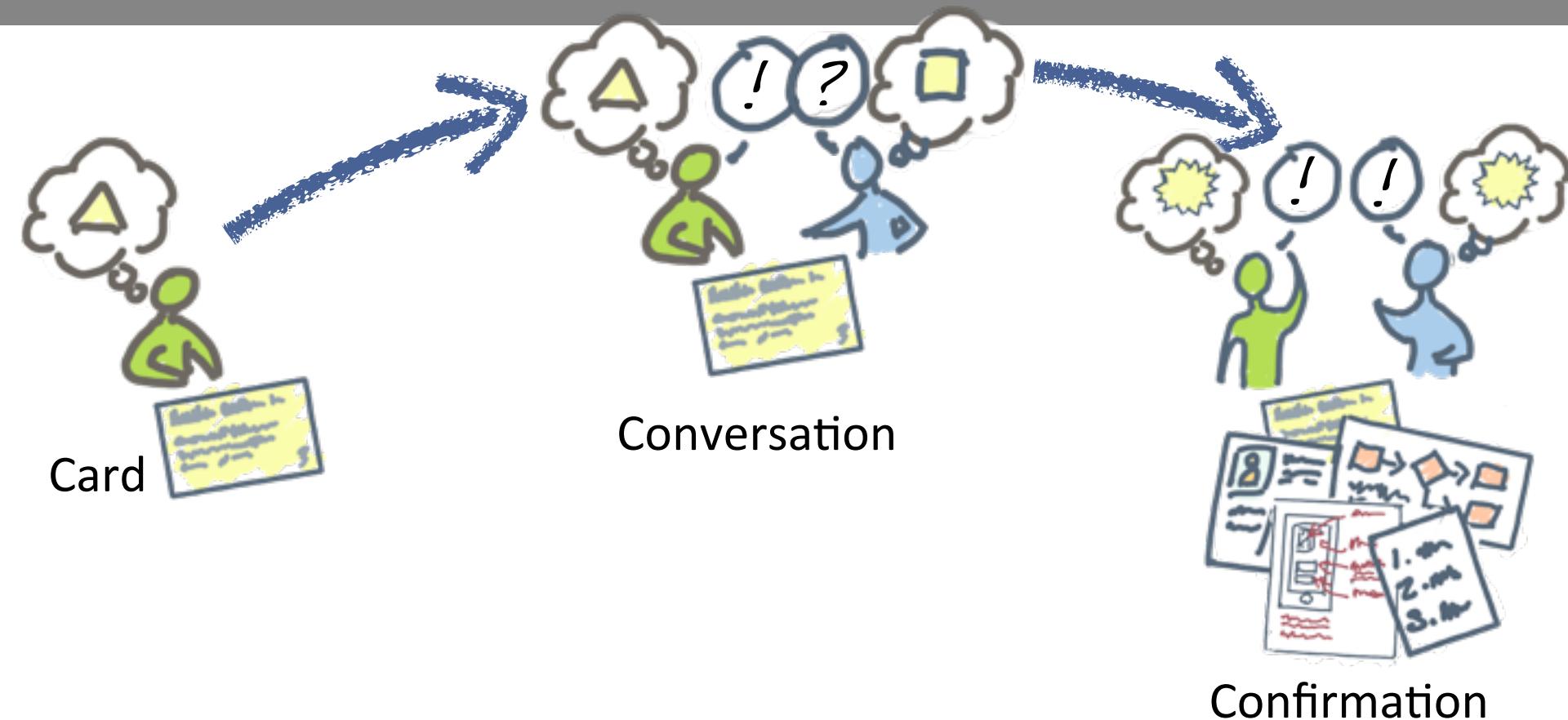


Rachel and her team at Connextra created a clever conversation starter



Focus discussion and collaboration around who will use the product and how they'll work “later,” after delivery

Stories have a simple lifecycle



* Ron Jeffries coined the 3 C's in
Extreme Programming Installed



If you replace a conversation with a document, you've stopped using stories

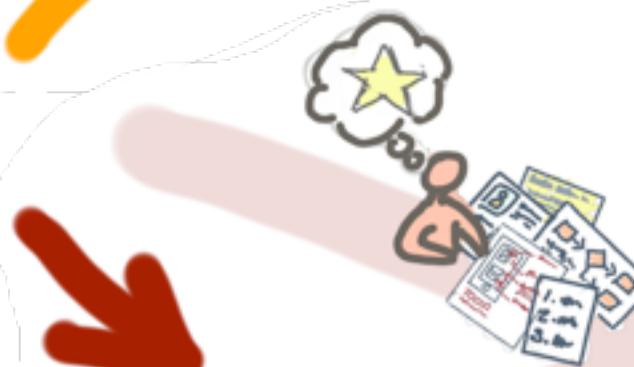
Start with
Shared Understanding



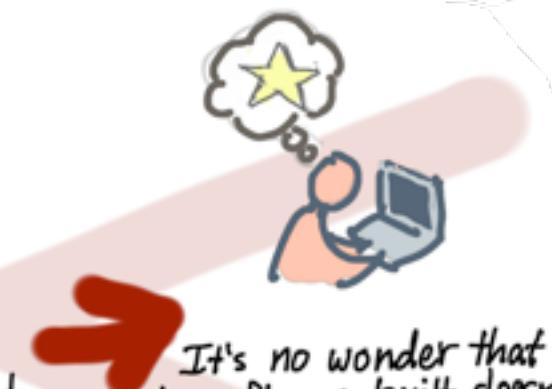
Spread Shared
Understanding with discussion



Build software equipped
with Shared Understanding



Reading documents
alone results in a different
understanding...



It's no wonder that
the software built doesn't
match our expectations



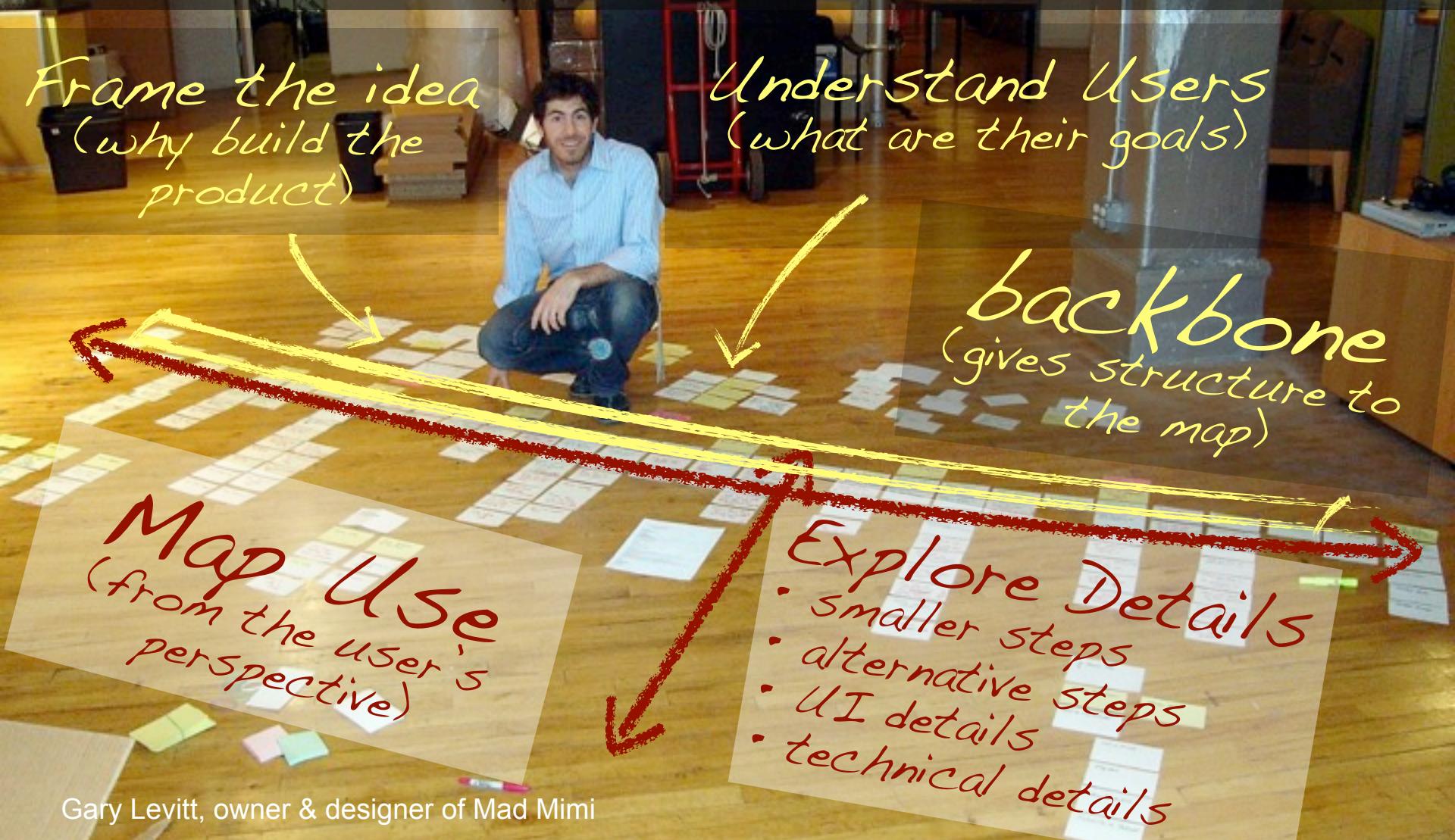
Stories aren't a
different way to write
requirements, they're
a different way to
work



It's easy to get
nowhere fast



A Story Map helps organize discussion about user's experience with our product



Gary ultimately built a successful product

The screenshot shows the TechCrunch website. At the top, there's a green 'TC' logo, a 'Got a tip? Let us know.' button, and a blue banner for 'Dropbox for Business'. Below the header, there are navigation links for News, TCTV, Events, and CrunchBase, along with social media sharing icons and a search bar. A sidebar on the left lists categories like email marketing, GoDaddy, Mad Mimi, Fundings & Exits, and Popular Posts. The main article headline is 'GoDaddy Acquires MailChimp Competitor Mad Mimi To Beef Up Its Email Marketing Service', posted by Sarah Perez on Aug 20, 2014. The article features a large image of a smartphone displaying a website. To the right of the article is an advertisement for Johnnie Walker.

Got a tip?
Let us know.

How to work together from anywhere:

Dropbox for Business
It's that simple
TRY IT FREE

News - TCTV - Events - CrunchBase

Follow Us

Search

DISRUPT: Disrupt SF begins next week. Don't miss out. Register today.

Updated AOL Privacy Policy and Terms of Service

email marketing

GoDaddy

Mad Mimi

Fundings & Exits

Popular Posts

Here's What We Know So Far About... a day ago

Apple Denies Any Breach Of Its Systems... 7 hours ago

Minuum Previews Its Size-Shifting... 5 days ago

Apple's New Spaceship HQ Doesn't... 2 days ago

3D-Printed 'Bump Key' Can Open... 6 days ago

GoDaddy Acquires MailChimp Competitor Mad Mimi To Beef Up Its Email Marketing Service

Posted Aug 20, 2014 by Sarah Perez (@sarahintampa)

20 Share 810 Shares 599 Tweet 553

Next Story

ADVERTISEMENT

KEEP WALKING. JOHNNIE WALKER. HUFFPOST LIVE

Welcome to The Next Step series presented by Johnnie Walker.

WATCH NOW

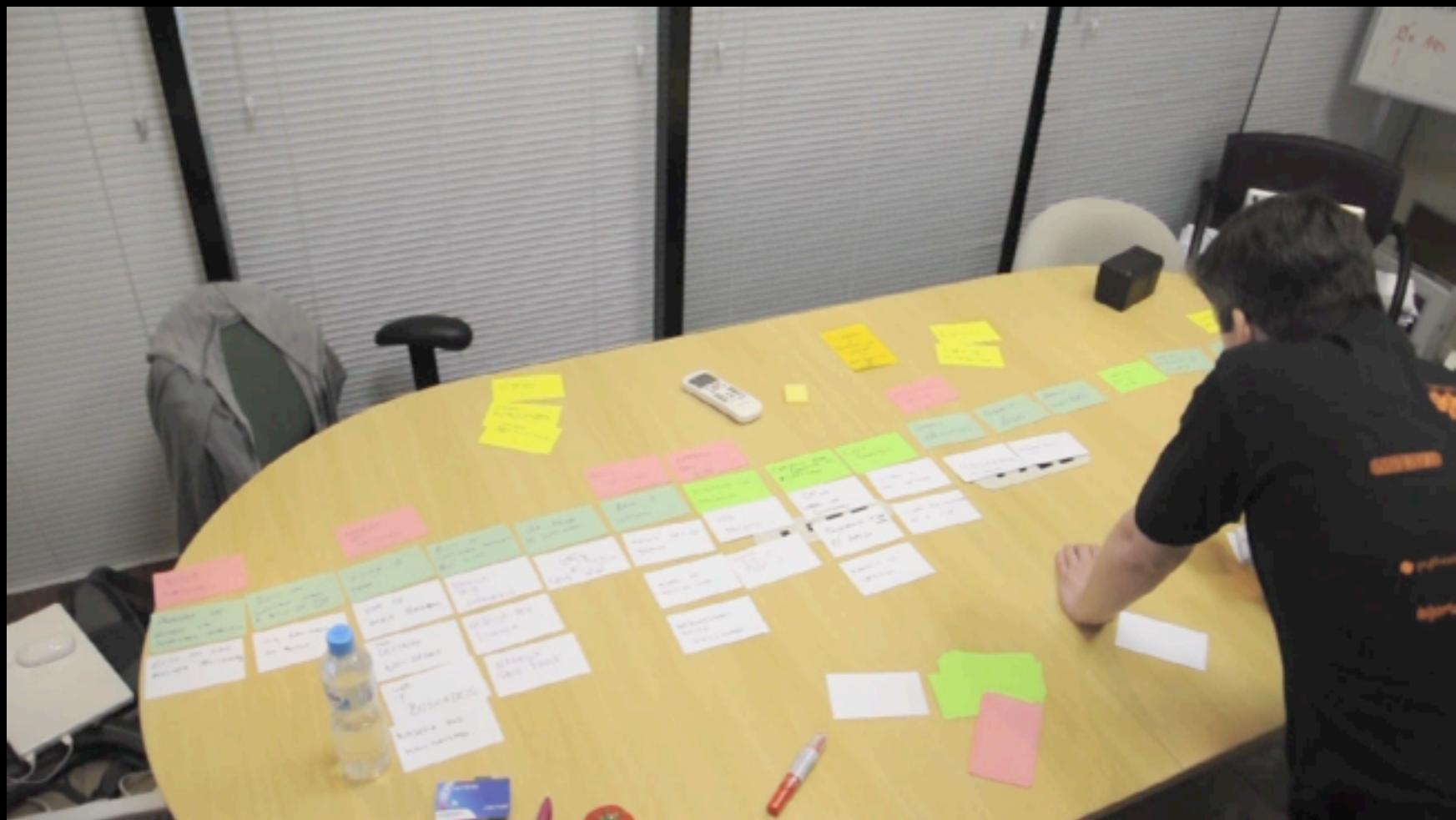
BY-NC-SA 53

GoDaddy has closed on the acquisition of MailChimp competitor **Mad Mimi**, the companies are announcing this morning. The move is meant to further expand the web hosting provider's product suite aimed at small businesses, while filling

A story map is a simple way to tell a story and break it down into parts



Build story maps in small collaborative groups



Use the map for continuous discussion



Discussions drive out more details, validate, and build shared understanding



Talking through the map with multiple users and subject matter experts helps test it for completeness

Use story maps to understand
your whole product or
feature's experience

Use mapping to break down
big stories without losing the
big picture



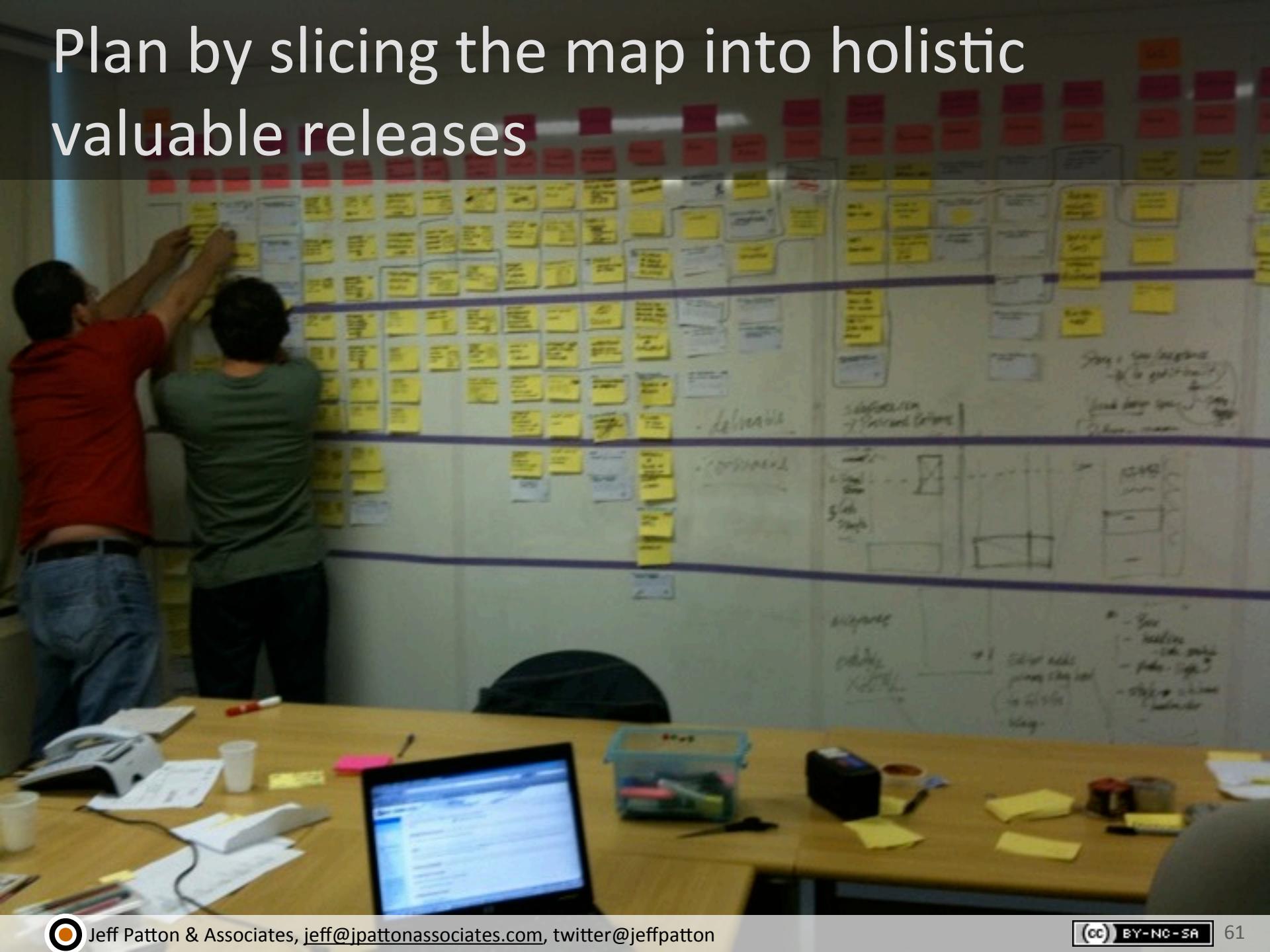
If I hear MVP one
more time, I'm going
to shoot myself



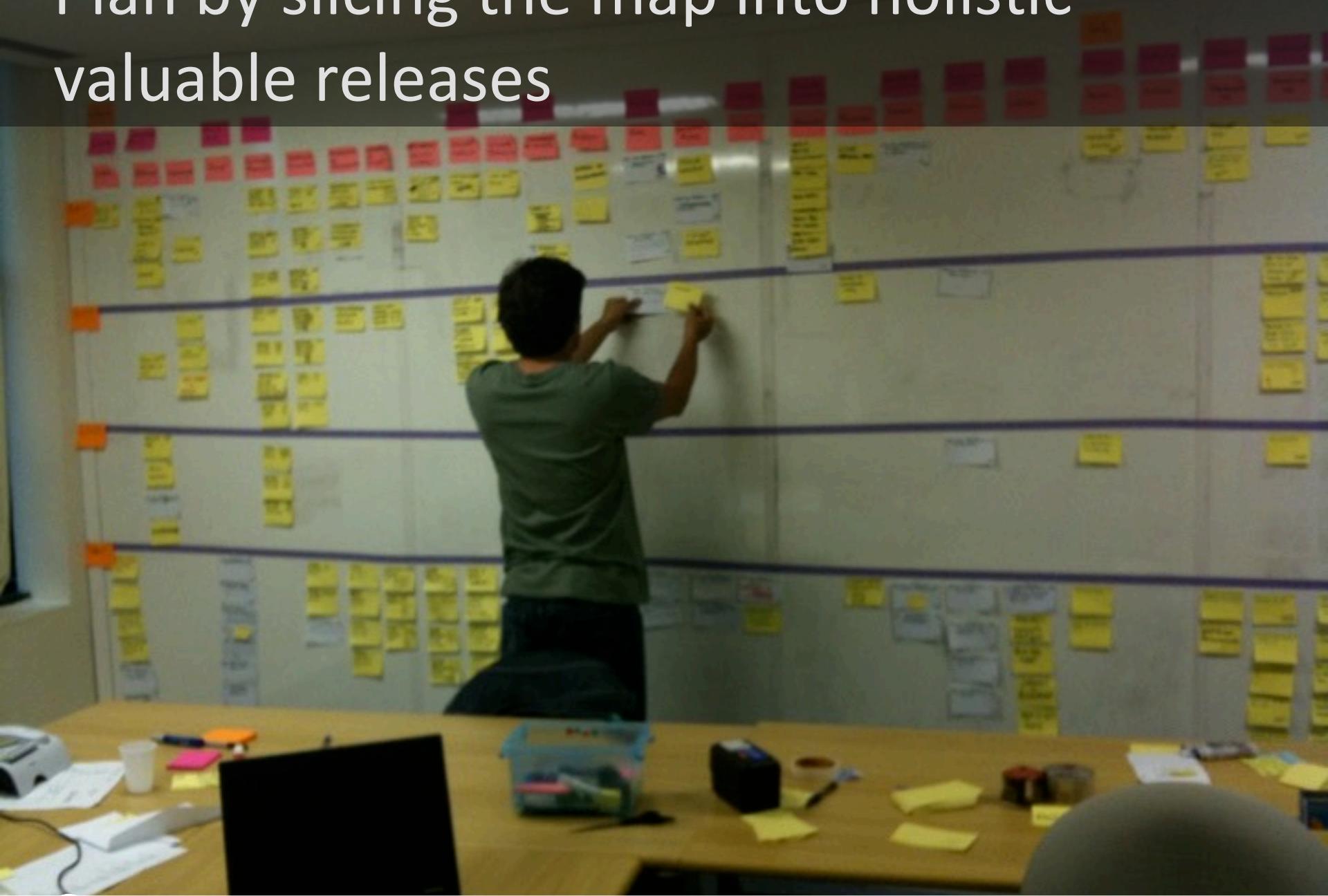
Plan by slicing the map into holistic valuable releases



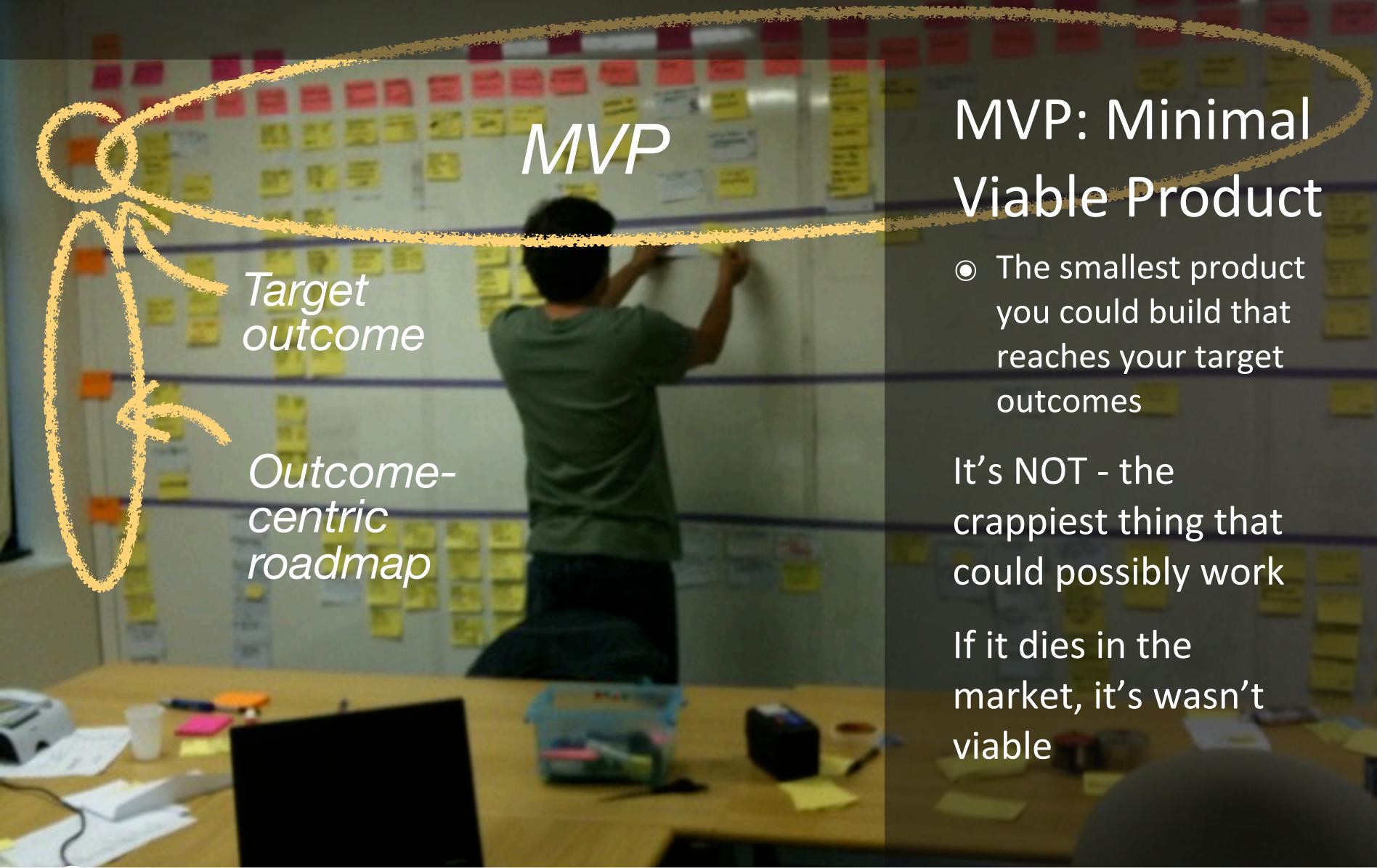
Plan by slicing the map into holistic valuable releases



Plan by slicing the map into holistic valuable releases



Your job is to build LESS software



But, how do you
know if you're
hypothesis is correct?

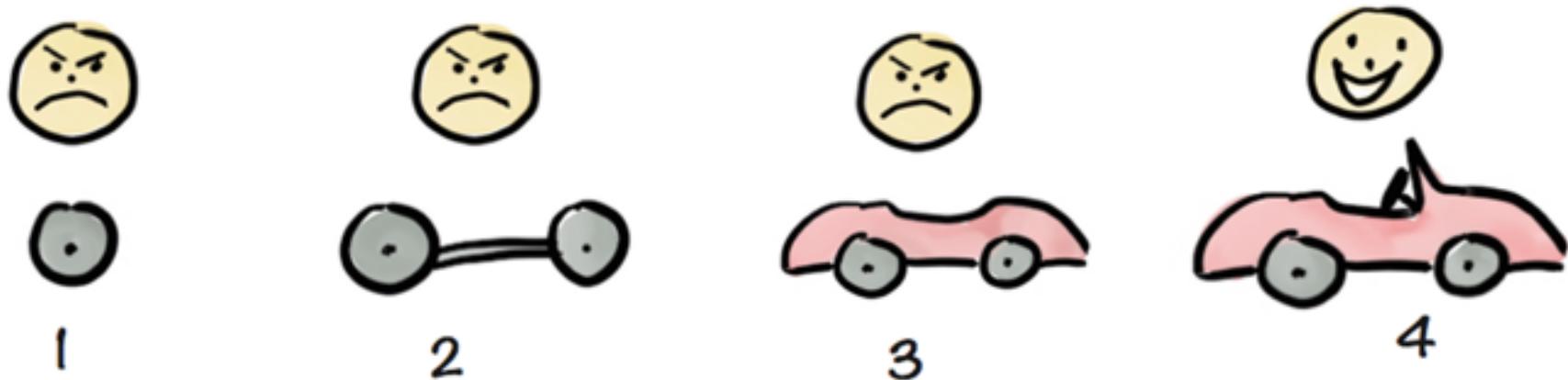


You don't



Delivering your hypothetical solution a piece at a time delays learning

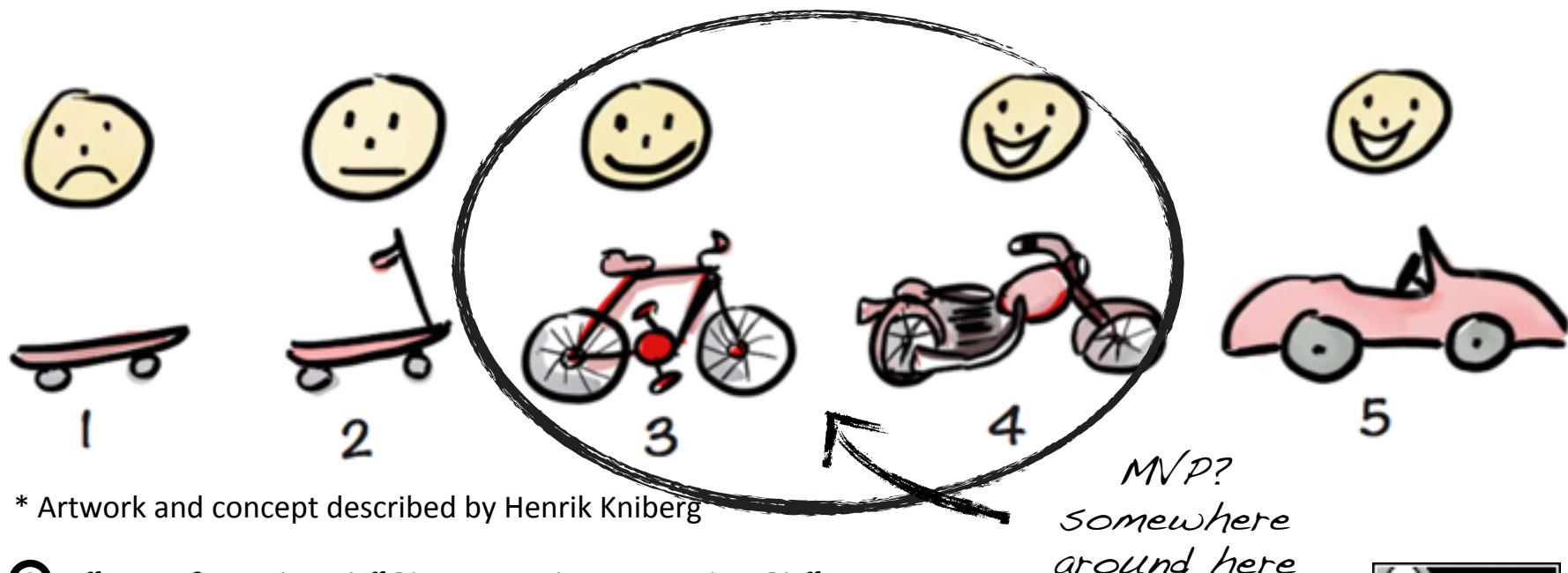
Hypothesis:



* Artwork and concept described by Henrik Kniberg

Deliver minimum viable product tests to a smaller audience to find what's really viable

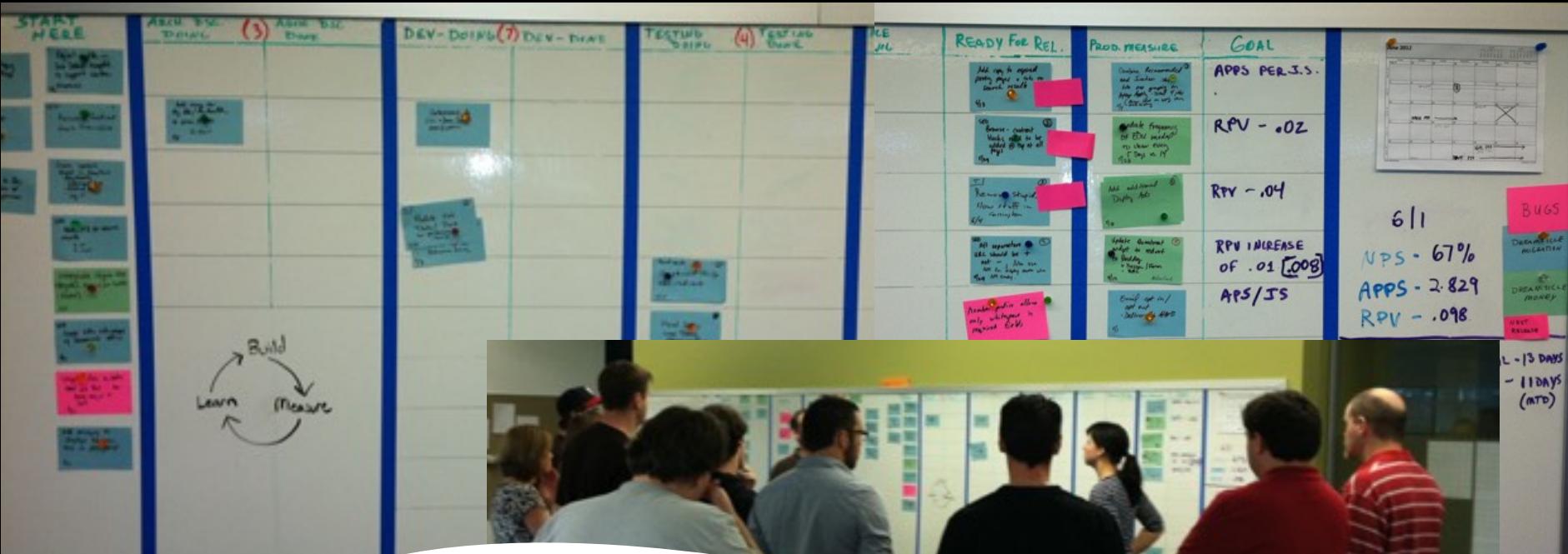
Hypothesis:



* Artwork and concept described by Henrik Kniberg

Eric has organized his backlog into a series of release slices





Explicit release step

Explicit measure step & metrics

Nothing leaves their board until there's been a discussion on what they've learned

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



You won't finish
on time



“accurate estimate” is
an oxymoron

To release benefit on a
schedule we'll need to
budget, and leverage
incremental and iterative
thinking
(What's the difference?)



“incrementing” builds a bit at a time



1



2



3



4



5



Incrementing calls for a fully formed idea.

And, doing it on time requires dead accurate estimation.

“iterating” and “incrementing” builds a rough 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

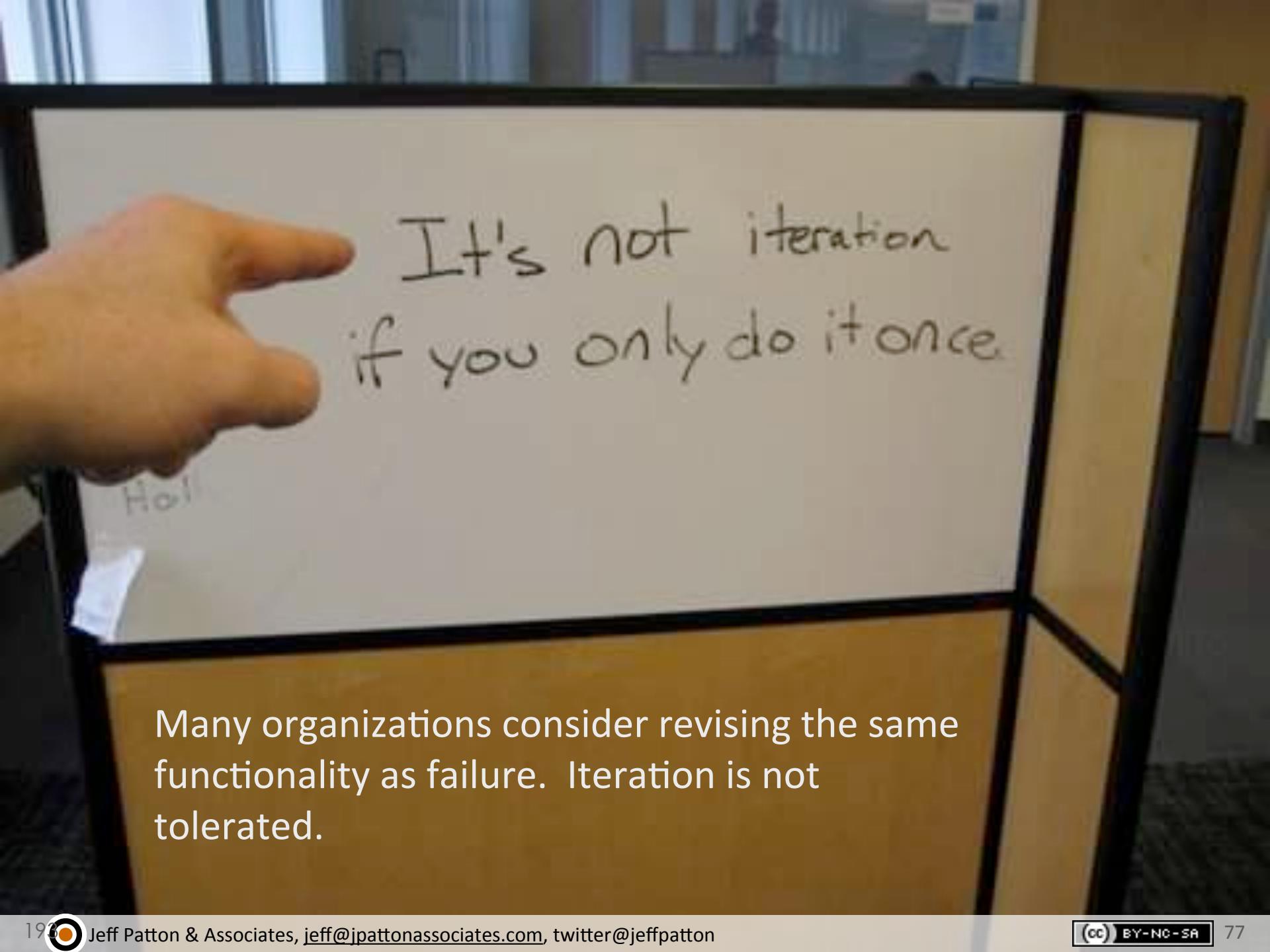


4



5

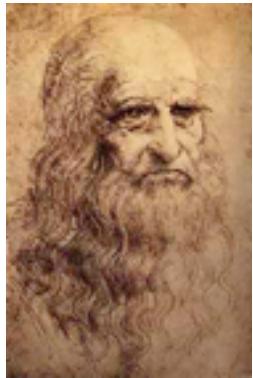




It's not iteration
if you only do it once

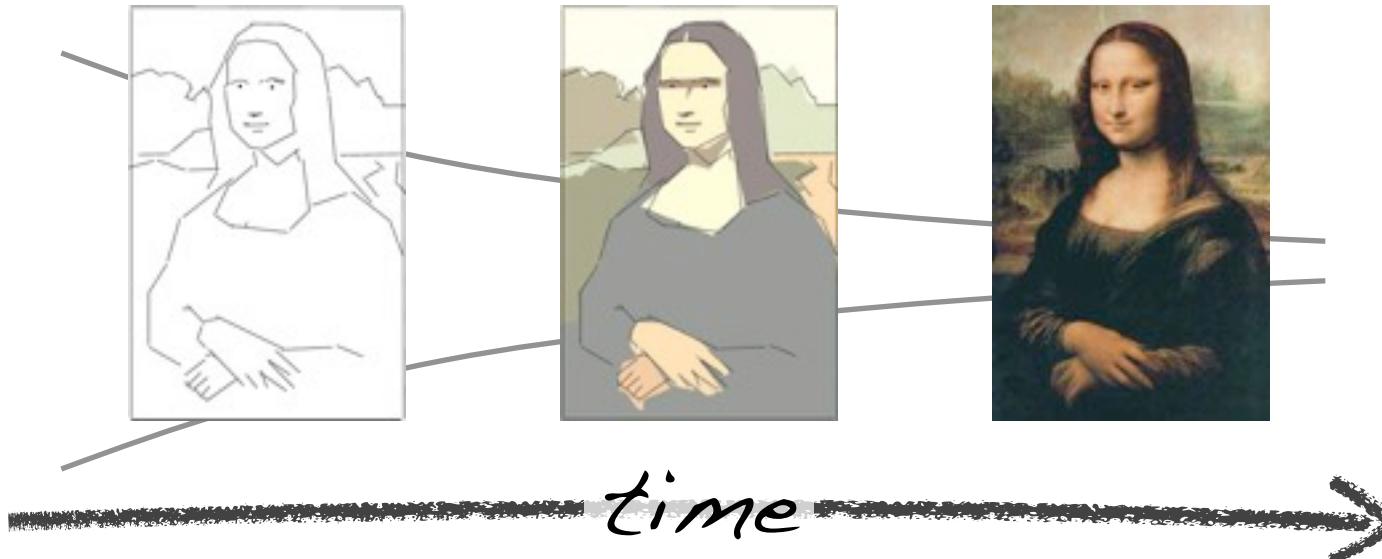
Many organizations consider revising the same functionality as failure. Iteration is not tolerated.

Work like an artist to envision and build the product holistically

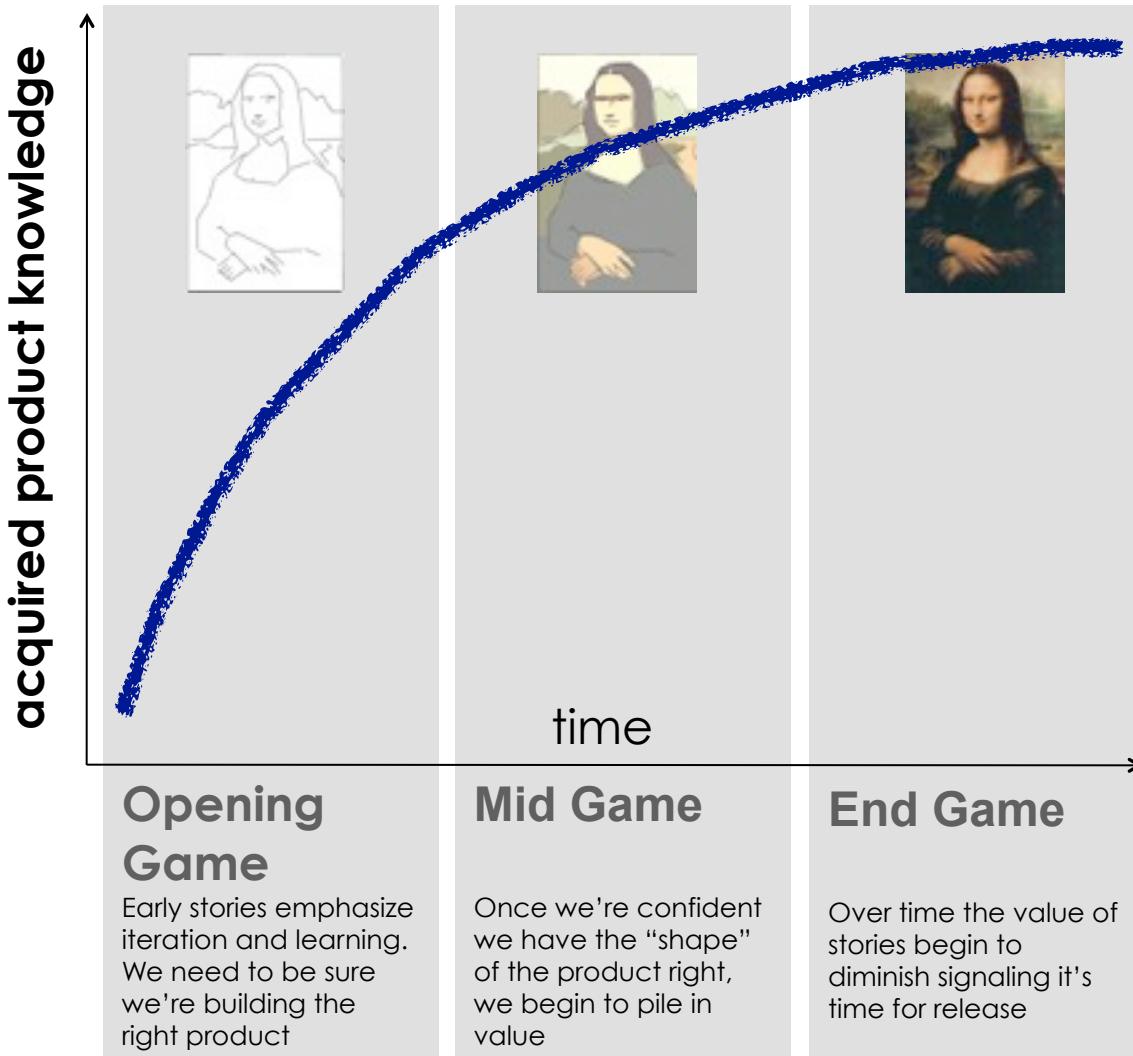


“Art is never finished,
only abandoned.”

-Leonardo DaVinci



Organize work to maximize learning

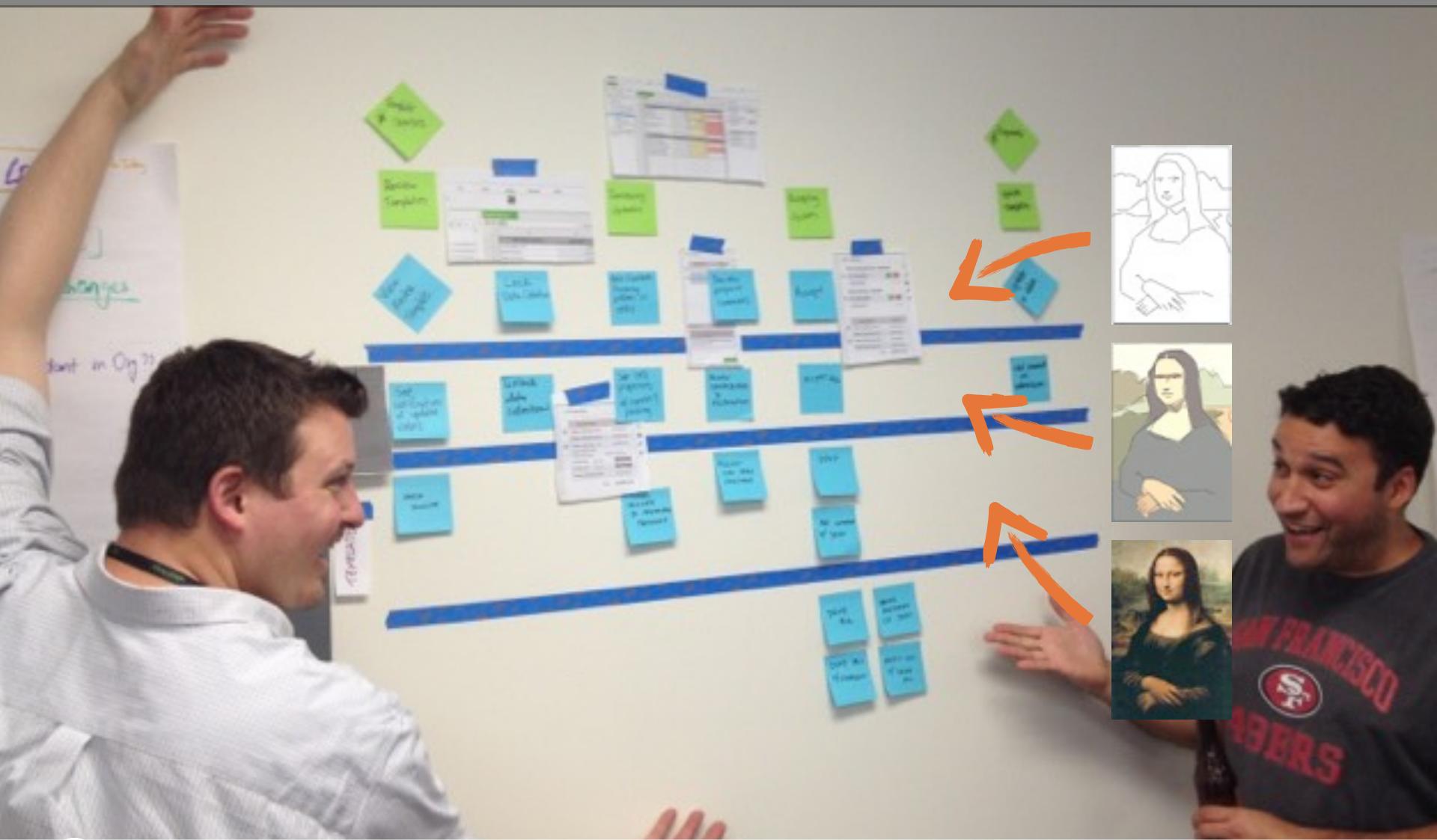


The inverse of risk is knowledge

Learning earlier about delivery risks helps us finish on time

Alistair Cockburn refers to cutting the small “polishing” stories as “trimming the tail.”

Use a story map to slice out a delivery strategy



Consider these four story splitting heuristics that build up quality

Bare Necessity

For the feature to be minimally demonstrable – but not releasable, what is the minimal functionality

Example: A form with only necessary fields and no validation

Capability & Flexibility

What would add the ability to perform the user task in different ways? Adding in sub tasks that are optionally performed?

Example: a form with optional fields, date lookup tools, input translation on dates

* Adapted from Gerard Meszaros' "Storyotypes"

Safety

What would make this feature safer to use? For both the user, and for the business paying for the software?

Example: input validation, enforcement of business rules such as credit card validation

Usability, Performance, Sex Appeal

What would make this feature easier to use? More desirable to use? Faster to use?

Example: auto-completion, sexy visual design, speed keys



Building up quality iteratively and incrementally ships the best product possible

sprint

4

A-

A

B

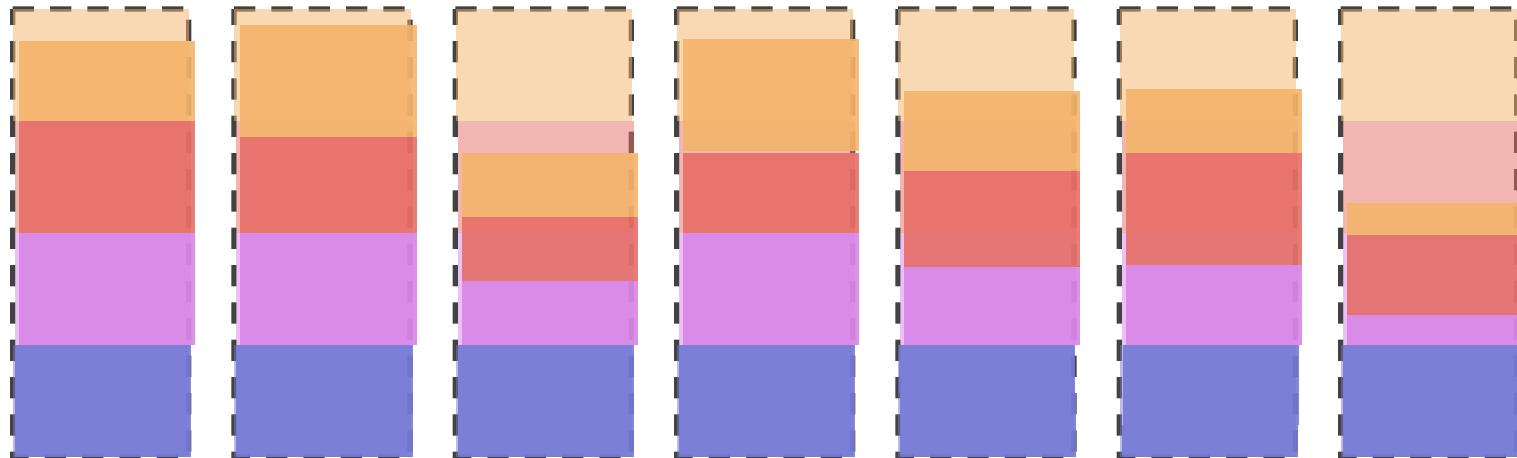
A

A-

A-

B-

user tasks to support



Product goal: **(in 4 sprints)** ship the best product possible



Product Owners must understand the delivery strategy that leads to a finished product



Sculpture at various stages of completion, Musée d'Orsay, Paris

Build up software
iteratively and
incrementally to release
the highest quality
possible on time

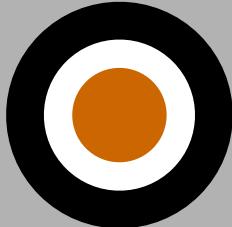
1. Tell stories, don't just write them
2. Use simple visualizations to anchor the stories you tell
3. Tell the whole story to find the parts that matter most
4. Think things through: minimize output, maximize outcome and impact
5. Build to minimum viable product tests to find what's minimum and viable in the market

Effective stories connect everyone to the purpose of your product



Story Mapping

discover the whole story



Jeff Patton
jeff@jpattonassociates.com
twitter: @jeffpatton

I wrote this
book!

