

Using Kanban (and lean thinking) to help dysfunctional agile teams



Jeff Patton
jpatton@acm.org
www.AgileProductDesign.com

Why me?

The hour will go something like this:

1. Simple explanation of Kanban
2. What's different about Kanban & common agile or scrum practice?
3. Common agile practice problems
4. Stories where adding Kanban-ish practice helped teams improve

and we'll watch two
cartoons...

and a documentary.

1. Kanban in 5 minutes

A Kanban is a sign



これがその看板
694×489 - 74k - jpg
gigazine.net
[類似の画像を探す](#)



【マクドナルドの空中】
299×368 - 12k - jpg
mochikaz.blogspot.com
[類似の画像を探す](#)



しかもこの看板
640×480 - 89k - jpg
candytower.com
[類似の画像を探す](#)



看板だそう
400×533 - 34k - jpg
gigazine.net
[類似の画像を探す](#)



ブティックのFRP造形
539×404 - 45k - jpg
charako.jp
[類似の画像を探す](#)



たんぼぼ保育園の
640×480 - 308k - jpg
oohito.com
[類似の画像を探す](#)



看板・のぼり
350×350 - 75k - jpg
at-imagine.com
[類似の画像を探す](#)



フラッグの看板
450×338 - 147k
blog.goo.ne.jp
[類似の画像を探す](#)



A型看板ス
314×508 - 92k - jpg
item.rakuten.co.jp
[類似の画像を探す](#)



変てこ看板
480×640 - 136k - jpg
stevie.ti-da.net



電柱看板
336×448 - 23k - jpg
yomeishu.co.jp
[類似の画像を探す](#)



【マクドナルドの空中】
354×400 - 21k - jpg
mochikaz.blogspot.com



宮古島にお越しの際は
500×375 - 107k - jpg
coralvegetable.ti-da.net



で、のような看板
1600×1200 - 577k - jpg
blog.presso-inn.com
[類似の画像を探す](#)



合成CGで看板
300×338 - 36k - jpg
grandswell.co.jp
[類似の画像を探す](#)



この新しく設置された
640×480 - 32k - jpg
start.hamazo.tv



富士看板
432×300 - 41k - gif
kanban.co.jp
[類似の画像を探す](#)



看板デザイン
300×338 - 30k - jpg
grandswell.co.jp
[類似の画像を探す](#)



A型看板の
663×375 - 45k - gif
akanban.signmall.jp
[類似の画像を探す](#)



看板はかん
740×440 - 63k - gif
kanban-ya.co.jp
[類似の画像を探す](#)



街角の看板
338×450 - 67k - jpg
mabillon.jugem.cc
[類似の画像を探す](#)

A Kanban card is a sign-card representing something of value



Limiting Kanban tokens helps prevent over-production

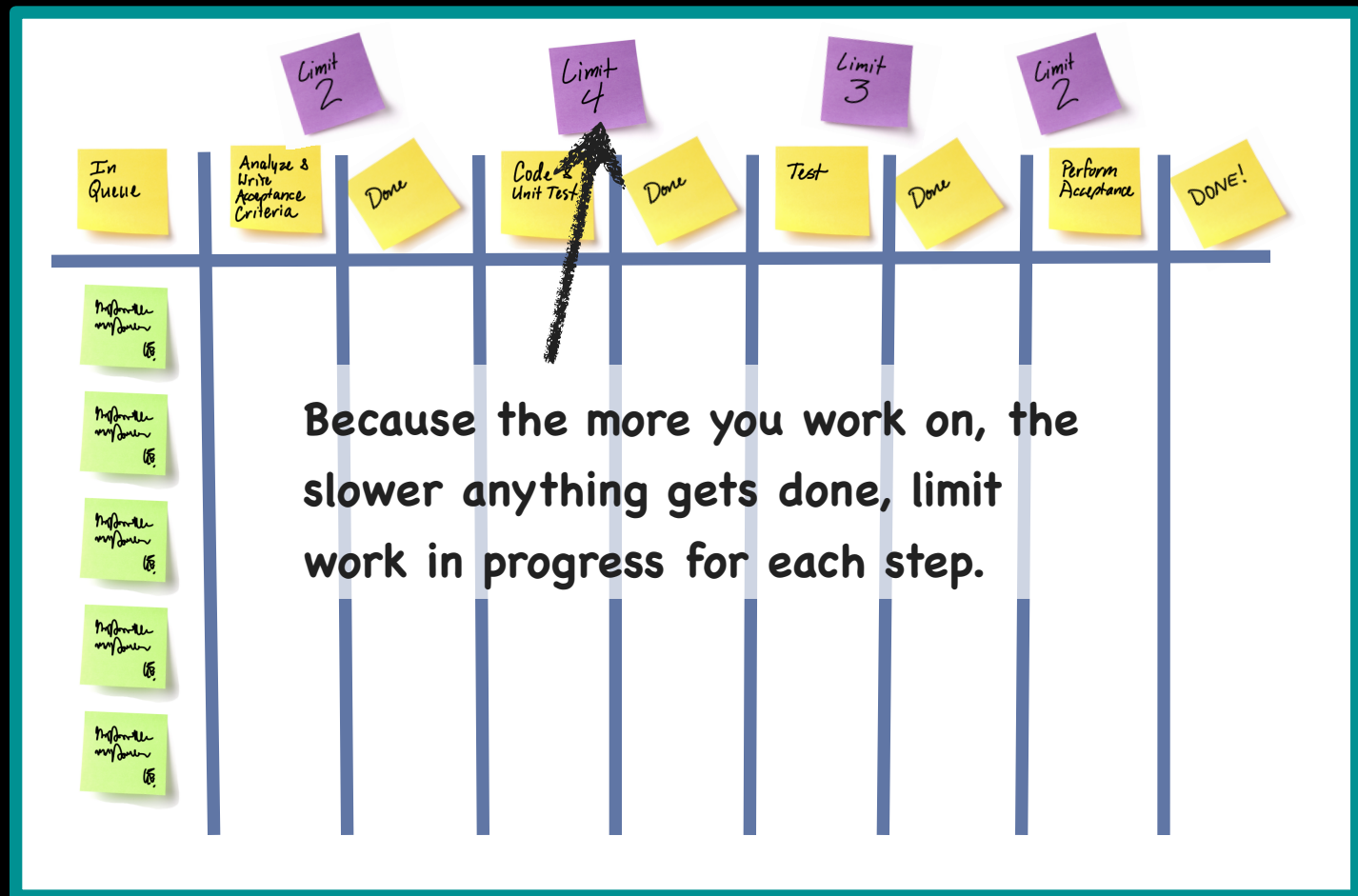


A Kanban board visualizes the flow of value

Our process for building things →

Load stories here →

As work progresses, move kanbans to show flow of value



2. Scrum & Kanban

What's different?

Nothing

(At least nothing needs to be)

Some Lean principles in play:

1. Use visual control so no problems are hidden
2. Level out the workload (heijunka)
3. Use the "pull" system to avoid overproduction (can mean developers stop working if testers can't keep up)
4. Create continuous process flow to bring problems to the surface (can mean no time-boxes)

Start by using Kanban
boards to visualize your
current process

“Inspect and adapt” as you
see fit

3. Some common problems getting going with agile

Disclaimer:

No one said agile was easy.

Having difficulty getting going
with agile only makes you
human.

Problem 1: QA isn't engaging

“How do you expect me to test everything when I don't see it till the last day of the sprint?”

“Why should I test if it's just going to change later?”

“How can I test without a detailed spec?”

“Tasks? I can write ‘test-it’ on a sticky if you want.”



Problem 2: Developers only focused on their own work

“I’m working on all these stories together. They’ll be finished by the end of the sprint - I’m sure.”

“I can’t break this story down any more than it already is.”

“I won’t know what the tasks are until I start doing it.”

“Why should I estimate if the release date isn’t going to change?”



Problem 3: Analysts and UI people are overwhelmed

“I was up till 2 am getting ready for the next sprint - and the team’s telling me I don’t have the information they need!”

“I can’t look at that story now... maybe tomorrow.”

“Everyone wants input on what I’m doing - but they don’t have time, and neither do I.”



4. Stories about adding Leanish, Kanbanish practices and how it helped people improve

Disclaimer:

The stories you're about to hear are true. Only the names have been changed to protect the innocent.

Skeptical testers

The team plans an iteration

The testers don't test till
development is done - or maybe
a couple days later

The iteration finishes with testing
not complete

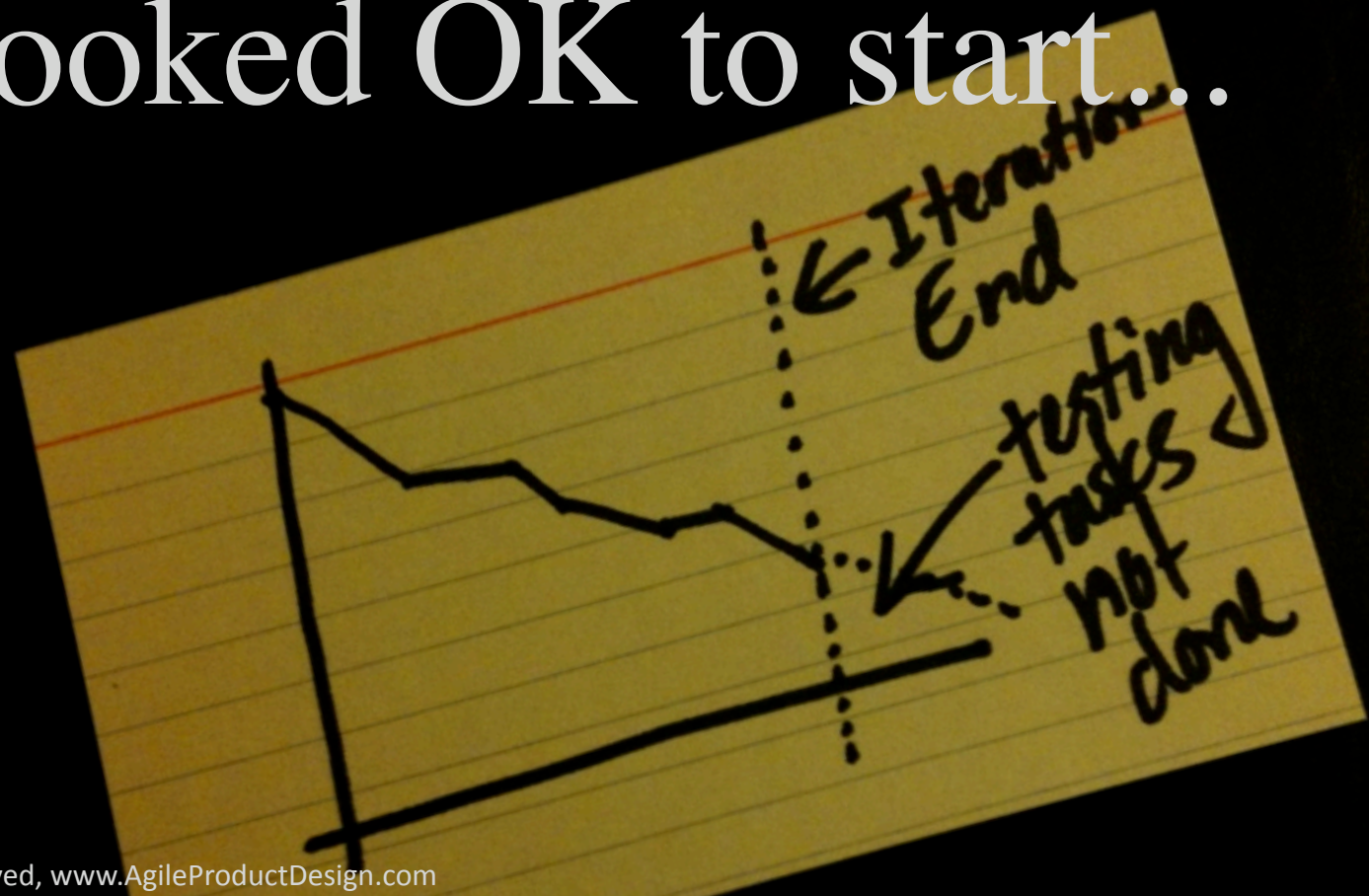
The team seems surprised

Scrum Master: “We’ll show the burndown chart so we can see we’re not getting done.”

Tester: “I have to write tasks?
Ok, I’ll write tasks that say ‘test the story’ then.”

The iteration ends.
Stories are still not done.

The burndown chart sorta
looked OK to start...



Everyone knew things
weren't going right, knew
what they should be
doing, we just couldn't
clearly see it.

Dr. Quest on controlling invisible things



Make it visible

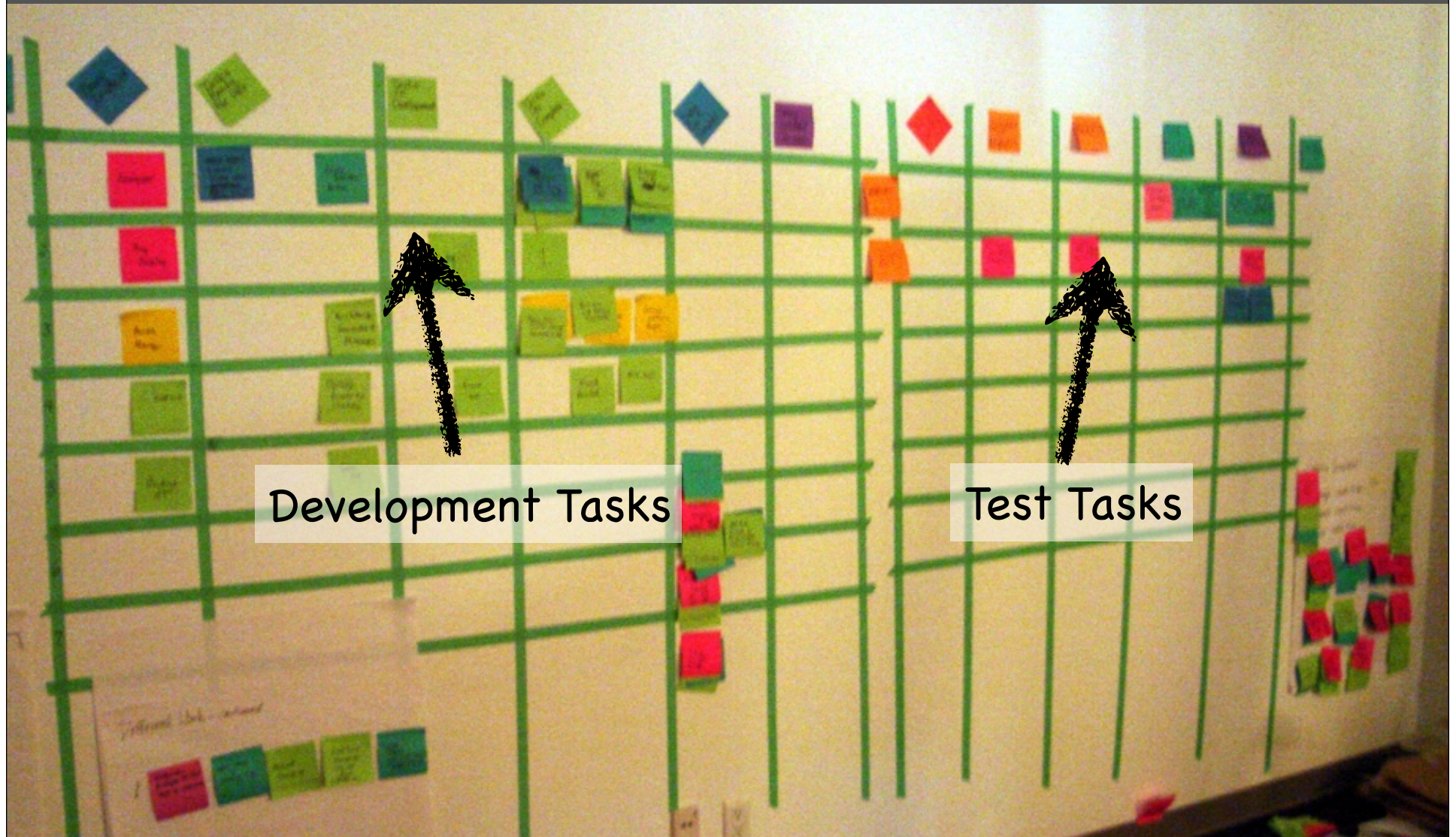
use simple visual controls so no
problems are hidden

Q: What information is hidden in
your work today?

Q: Does not knowing it hurt your
team's ability to be effective?

Coach: “I’ve built a task wall. Let’s load this iterations work onto it.”

Test tasks are clearly visible (so no problems are hidden)



Test tasks increase in quantity and sophistication



Developers begin to lean
out and collaborate

Developers lower their work in progress and working with testers create “testable tasks”

Testable Tasks

Reduced WIP!

Developer routinely collaborate with testers to get work moving



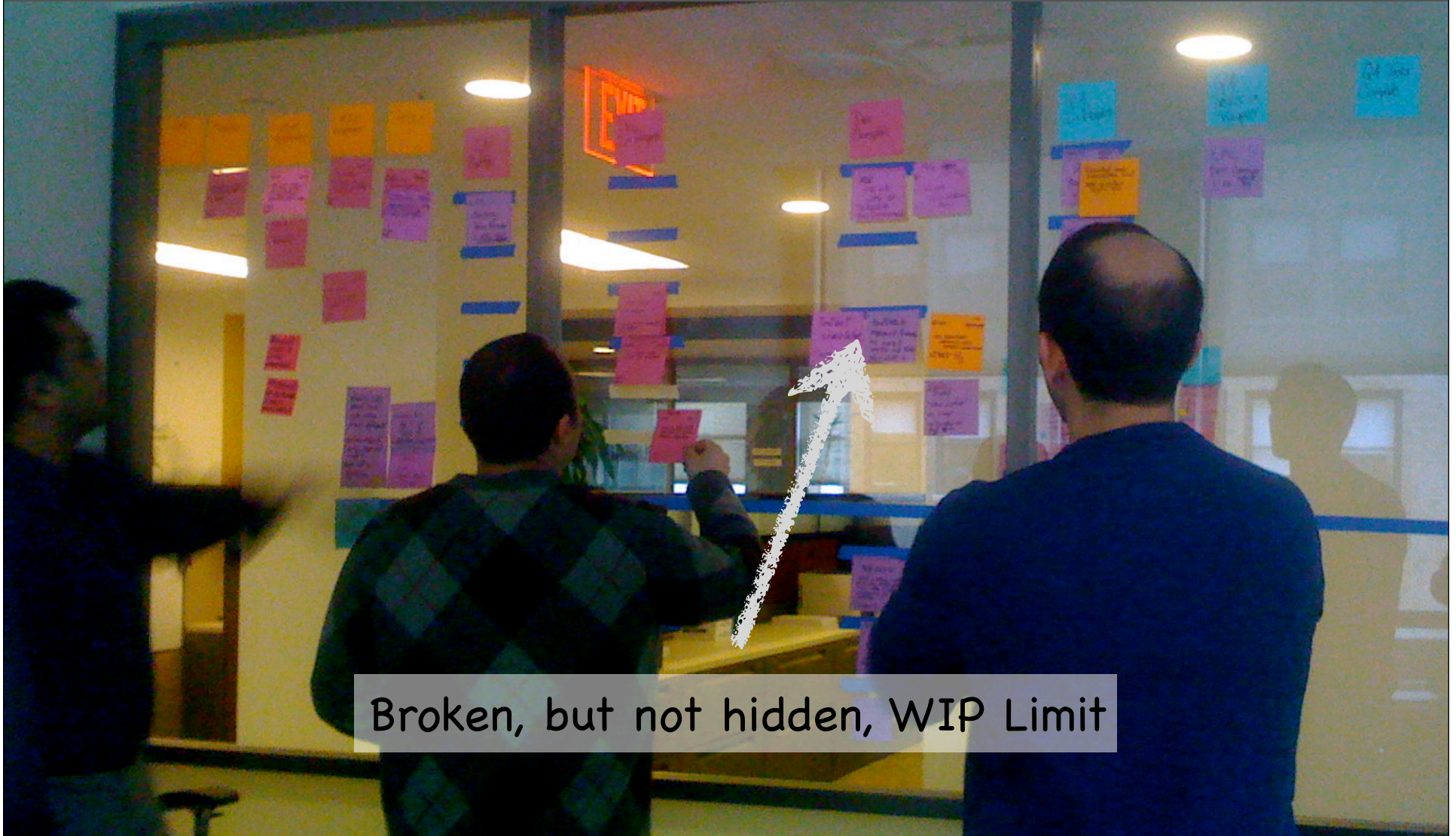
Meanwhile, the C++
developers boycott agile
outright

The agree to deliver every six
weeks into production

They commit to specific features
to delivery

But estimation and those silly
user stories are out

The agree to make their work visible, and build a kanban board in their team room



Broken, but not hidden, WIP Limit

Analysts and UI people
begin to visualize their
work

UI Designers visualize their workflow and work in progress



Someone's really overloaded!

Standup meetings take place where all the work is visible



Discovery work

Delivery

As the team draws near release, iteration boundaries drop, WIP limits drop, and workflow simplifies



At the end of the day, the
team didn't focus on
doing agile right, or doing
kanban right

The team focused on
collaborating, working
more effectively, and
delivering valuable
working software

Big benefits from informative Kanban boards:

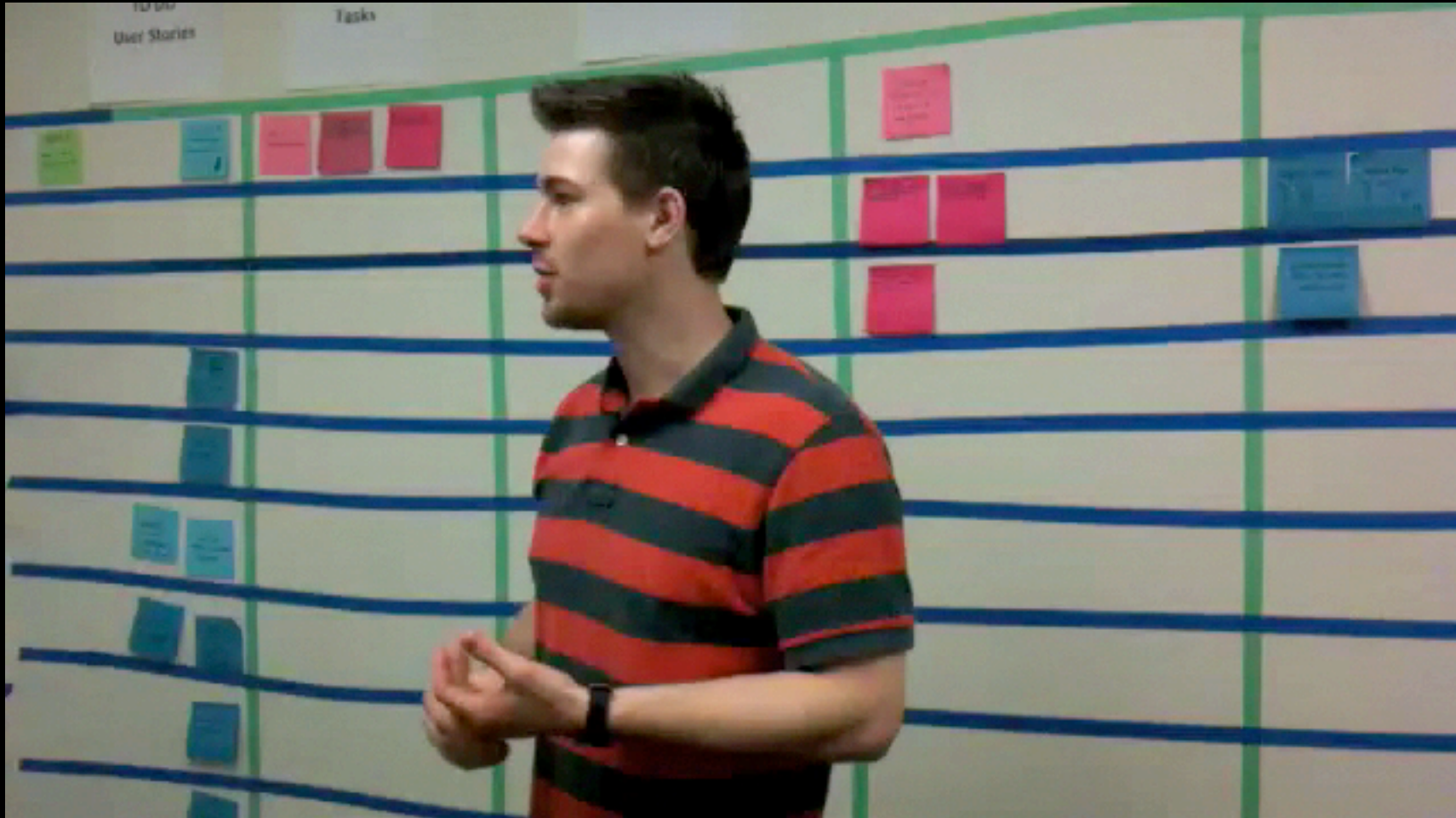
1. Simple process
visualization

2. Managing flow

Keep your iteration boundaries if you want

Let you're WIP run wild - you'll see and feel it

Eric describes his team's process today - listen for flow and pull



My favorite Kanban qualites

1. Simple process visualization

2. Managing flow

Using Kanban (and lean thinking) to help dysfunctional agile teams



Jeff Patton
jpatton@acm.org
www.AgileProductDesign.com