

Why your Agile rollout is failing

Dan North

DRW

Hi, I'm Dan

I'll be your consultant for today

May I see your watch please?

Yes, it's definitely Just After Lunch

Here is my invoice

Ok, let's cut to the chase

Your Agile rollout isn't going so well

You've asked someone to come in and take a look around, and make recommendations

If you don't tell anyone I didn't actually come in it can be our little secret...

So here's what I found! **cough**

The issues cluster into :

Kick-off

Planning

Finance

Communications

Change

Kick-off

You held a Big Bang launch

You said

We're going Agile!

and

This is for our customers / the Business

but you forgot to tell the development teams
why they should care

I recommend...

Win their hearts and minds

Explain: *what's in it for me?*

and: *why should I care?*

and: *what will “good” look like?*

Kick-off

You bought commodity training

You (or your boss) wanted Certification

but you only trained the project managers

who are now Certified \$crumMa\$ters

and who are a little nervous about their jobs

I recommend...

Understand how learning works
...and provide introductory recipes

Understand how change works
...and be prepared for it

then provide as-needed coaching

“75% of uncoached Scrum pilots will fail”

– Ken Schwaber

I recommend...

The Dreyfus learning model

Novice

Advanced Beginner

Competent

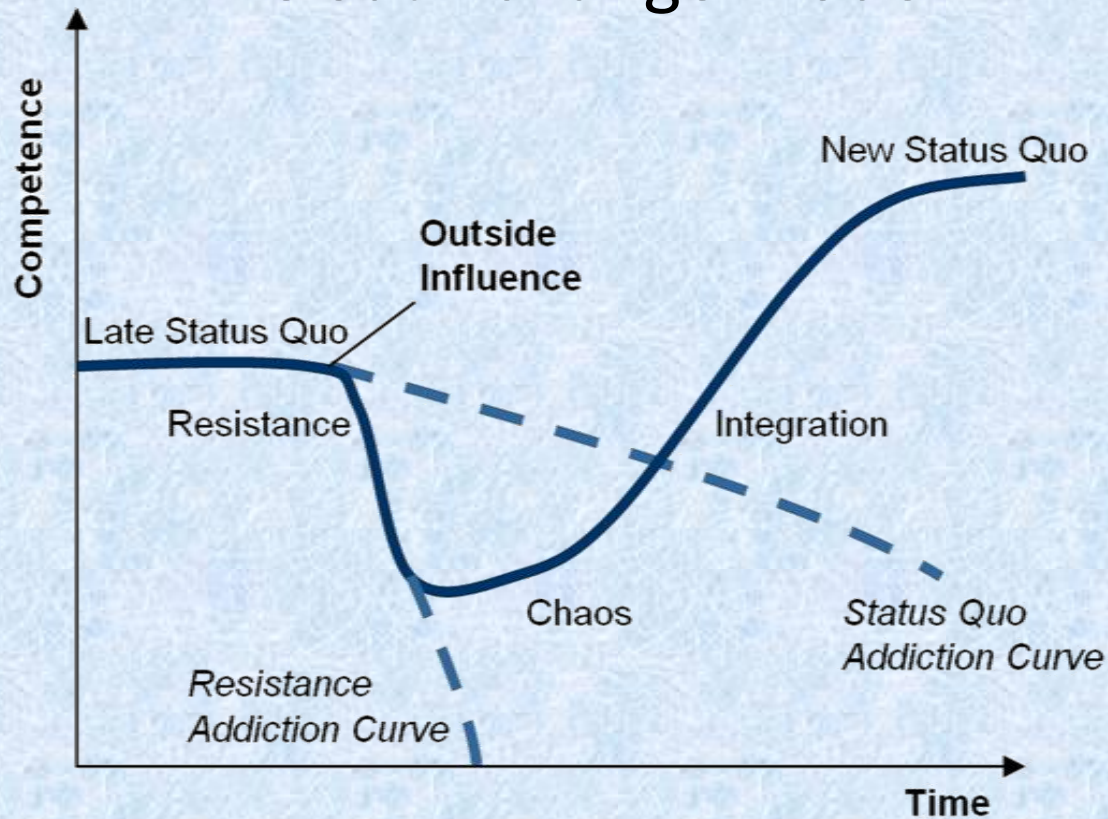
...time passes...

Proficient

Expert

I recommend...

The Satir change model



Kick-off

You started with planning and tracking

...but then you stopped there

You are suffering from *Scrumitis*

“Done” doesn’t actually mean done

...because you don’t have the technical discipline

...or a sufficient level of automated verification

I recommend...

Get a build!

“Come back when you have CI” – Ross Pettit

Understand your Path to Production

Automate your tests

and introduce “assumption tests” on legacy code

Planning

Your guys just don't get agile planning

They just turned the Gantt chart 90°

...the Task List becomes the Product Backlog

...and Tasks become Stories

Oh, and they don't know how to manage scope

What are they trying to deliver anyway?

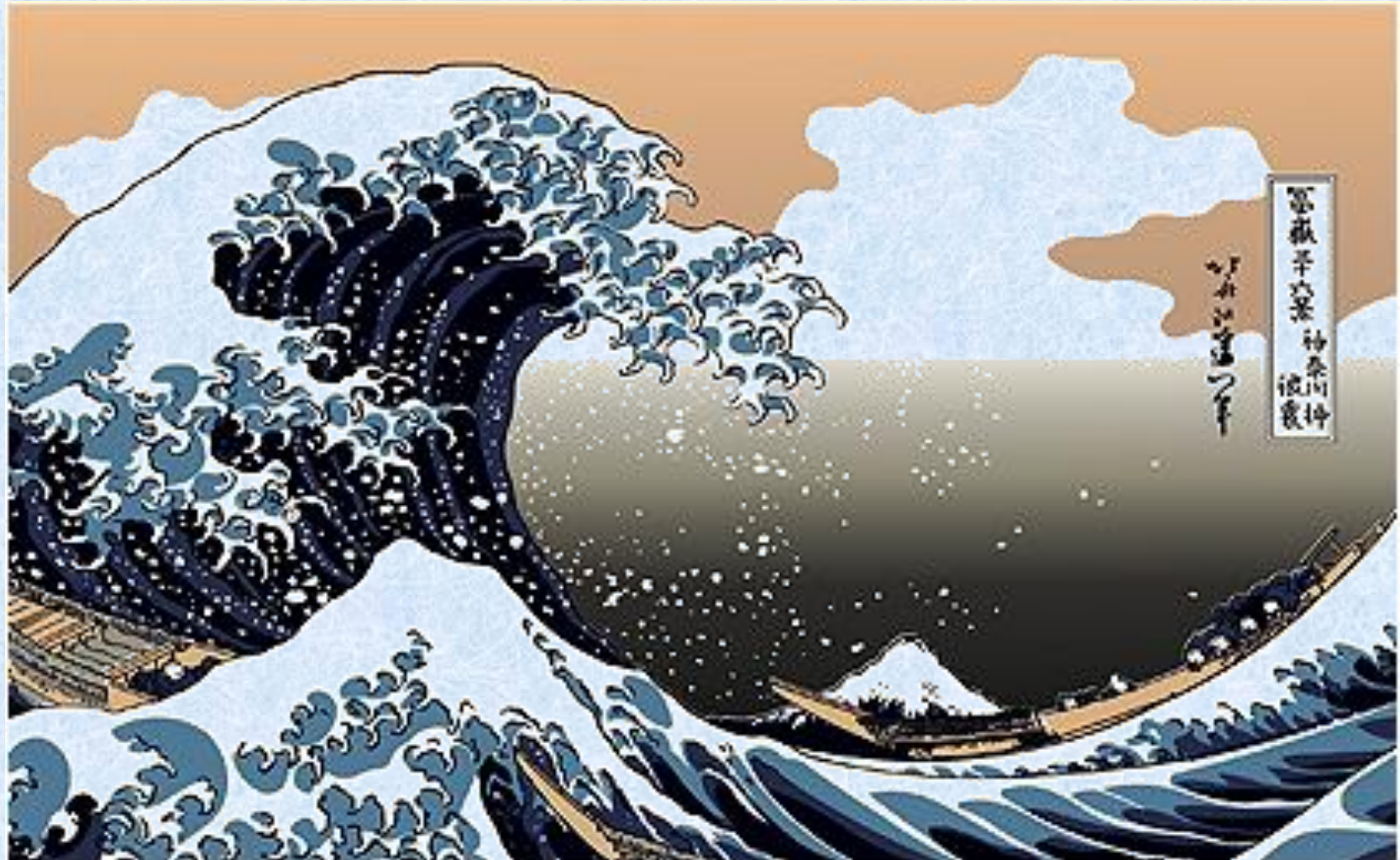
And when will the *project* be “done done”?

I recommend...

Understand *Deliberate Discovery*
along three axes

Understand *Rolling Wave Planning*
there is no spoon / MSL / Product Backlog

I recommend...



You think story points are useful

Effort != results

Activity != results

Results == results!

I recommend...

Have consistently small stories

Break features into stories

Get good at stories

The way to write good stories...

Then each story size is 1 Story

Your sprints are too long

4-6 weeks? You're doing "Waterscrum"
really just time-sliced mini-waterfalls

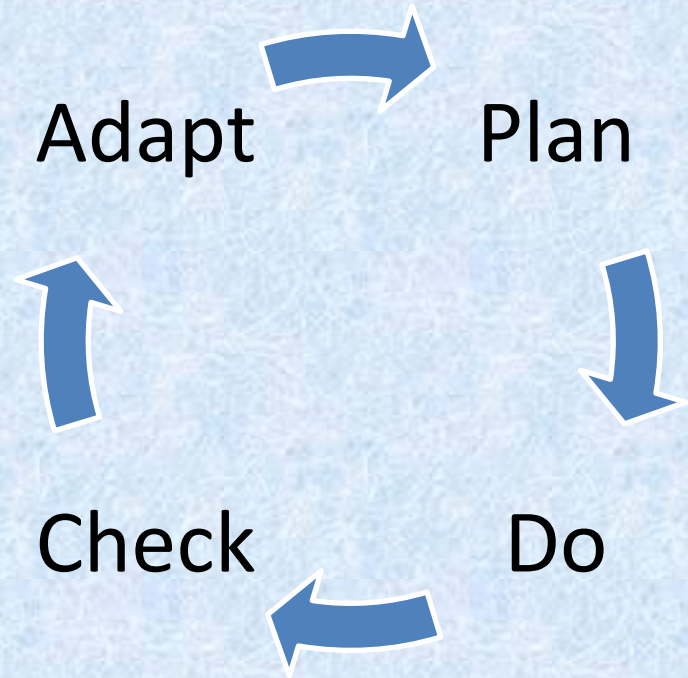
All the downsides of both Agile and Waterfall
your teams are white-water rafting!

I recommend...

Go to one week sprints

Identify what hurts

“We should fix that”



Finance

You “have an investment” in tooling

It was fit for purpose when you bought it

Hanging on to it will systemically hold you back

I recommend...

Spent money is spent!

Think of the impact on effectiveness

Make the cost of waste visible – in \$\$ and time
including opportunity costs
including impact on morale

You “have an investment” in infrastructure

You keep repurposing crappy old hardware

Enormous effort spent keeping it alive

I recommend...

Understand the waste implicit in repurposing

“You can get a petabyte of storage for about the cost of a decent developer (~\$100k)”

– Dave Thomas, 2009

“We could replace this with a cluster of iPods”

– Dante Briones, 2005

Communications

You said delivery wouldn't be impacted

You told the Business you're Going Agile

They said: *Sure, just don't let it impact any dates*

You said: *Ok, it won't!*

I recommend...

Build trust with the Business

Based on evidence

Demonstrate transparency

Demonstrate quick wins – celebrate success

Start small, get buy-in for more ambitious stuff

Operations didn't get the memo

Neither did your suppliers

The CM team aren't prepared for this

They can't cope with frequent releases

I recommend...

Engage them and explain Agile from *their* point of view

How will we engage them differently?

How will it help them?

Corporate didn't get the memo

The waterfall review gates are painful

HR don't know to change their hiring approach

All the KPIs are systemically resistant

effort-based

activity-based

I recommend...

Have a PR offensive!

Understand the intent of the review gates
and map them to a sane world

Help them rethink their KPIs
and remember, you will get what you measure

Change

Change

PMs won't pay for strategic change

“Not on my project budget”

Timescales too short to feel the benefit

They don't want to invest in technical practices

TDD, CI, etc.

I recommend...

Make PMs responsible for system lifecycle

Not just “this project”

Integrate with BAU

Make small changes frequently

Prove them, bed them in

Have a strategic change team

Give them teeth *and money*

Change

Architects don't trust the teams

Code review is a bottleneck
so is design

“Architecture by edict”

Blueprints and frameworks

Best Practices

I recommend...

Architects as consultants into teams

Incentivise by how much they share

Also applies to

Security

Systems Test and NFRs

Operations

Change

Finally (!) manual testing is hurting

and you off-shored it

to “save costs”

I recommend...

Pay for testing out of the delivery budget

Integrate testers into teams

Emphasis preventing rather than detecting

Invest in automated testing

Concluding thoughts

Agile adoption is about change and learning

Take time to understand change and learning

Lasting change requires Systems Thinking

You can't just "Go Agile"

"A bad system will beat a good person every time" – W. Edwards Deming

Thank you

dnorth@drwuk.com

[@tastapod](#)

<http://dannorth.net>

<http://jbehave.org>

<http://rspec.info>