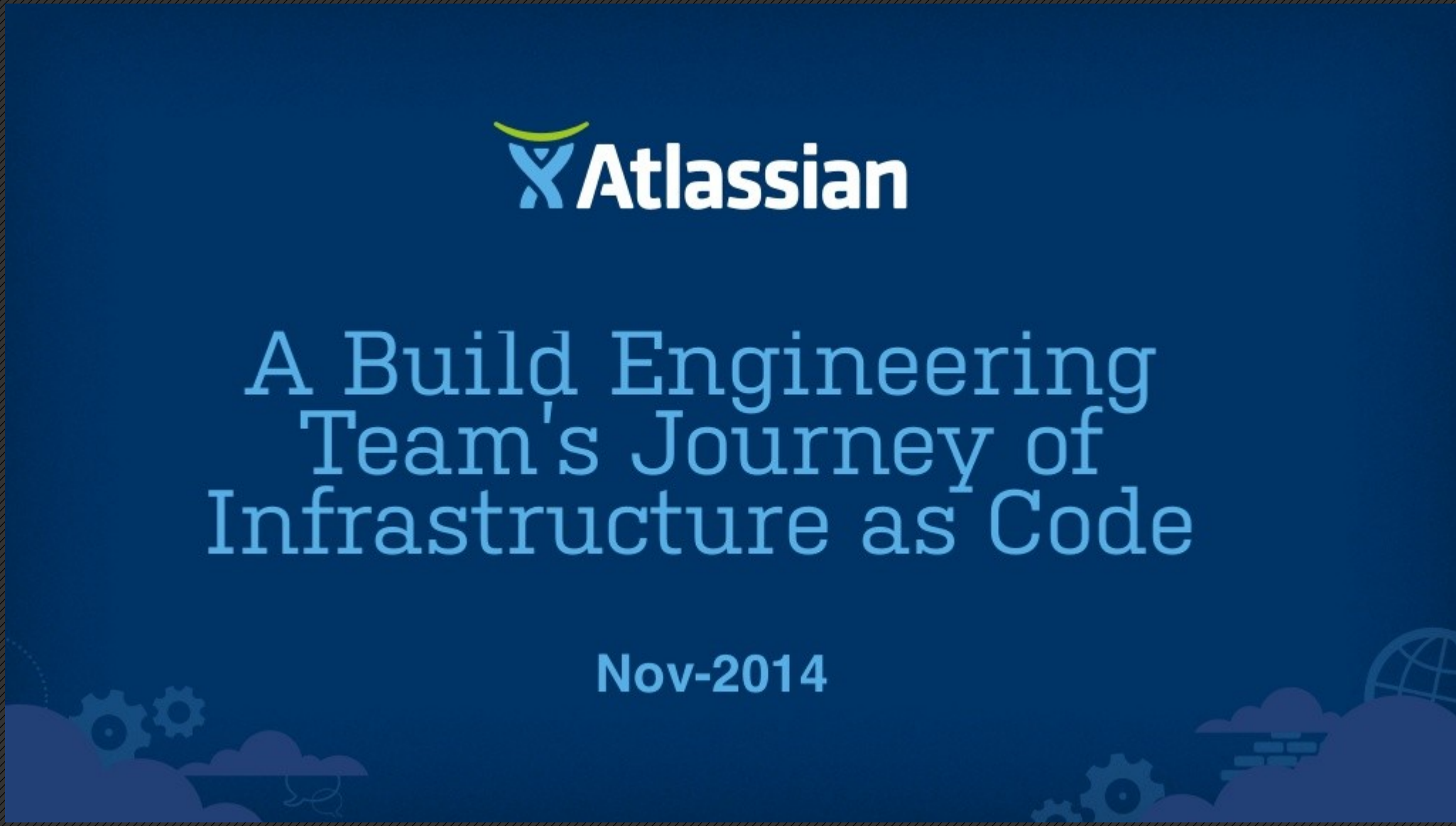
It's not the tools

"musing, ranting, and vain reflecting
on the relationship between IT operations and their tools."

Sydney devops meetup
Jan 15 2015

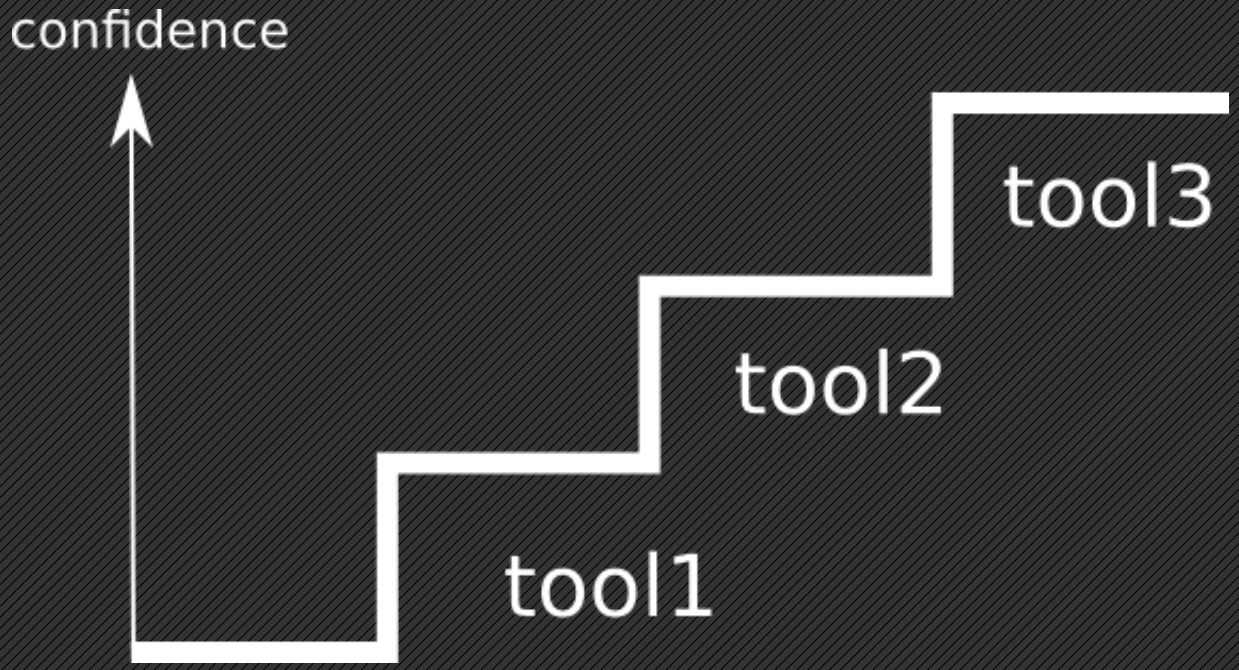
Gildas Le Nadan
@endemics

It all starts with a great reference



https://www.youtube.com/watch?v=FXdUxUdDhHM

My gripe?
A narrative around the pattern:



Reinforcing
a
MYTH

“For a given problem,
there is an off-the-shelf tool to
solve it”

(and I don't need to think)

WRONG

Why is that a problem?

(not so) Hidden costs of tools

Learn
Deploy
Integrate
Backup
Update
Support

It's probably even worse when it's a shiny new tool

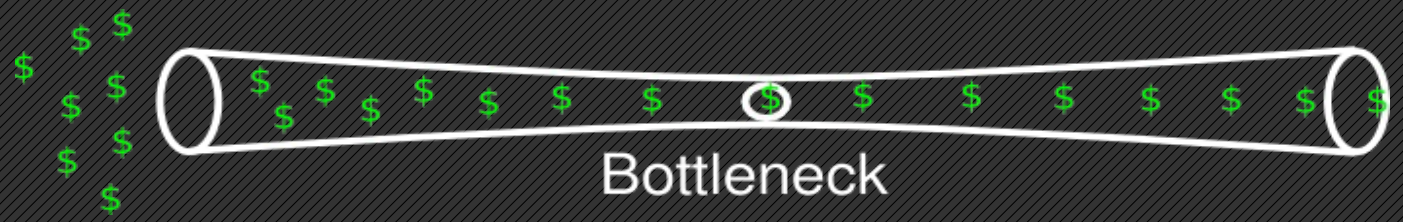
(Who said Docker?)

Some will even search for problems that match a tool

A **bold** statement?

“As IT Ops we're not here to play with new toys but to support the business.”

There is always a bottleneck in the value chain



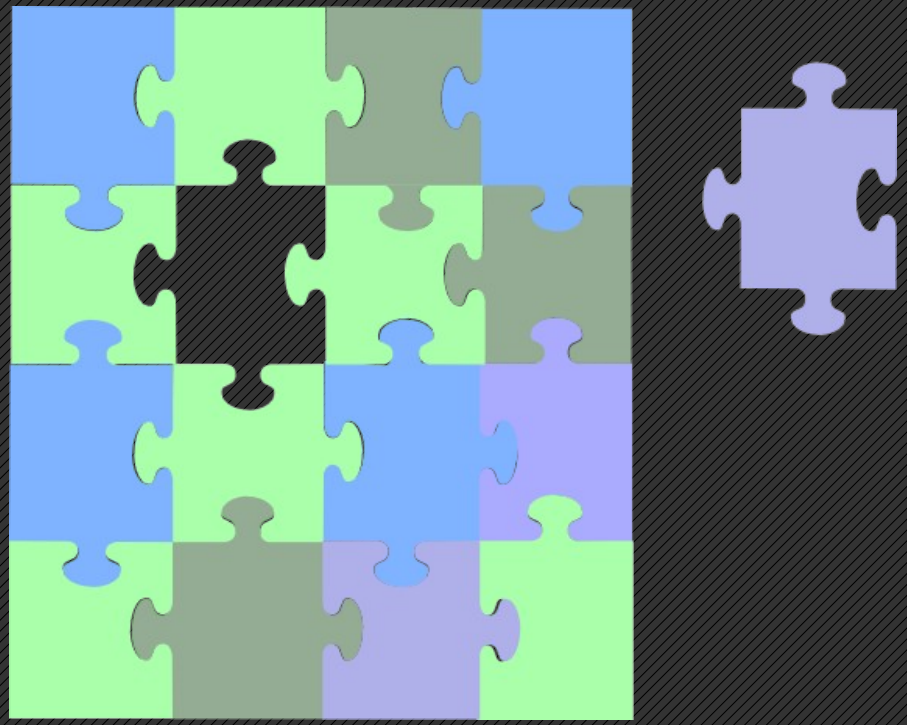
Be awesome

Don't be
the bottleneck
a cost center
a Nay sayer

Even the best tool won't cover all your use cases

So you'll have more of them

How most people view adding a new tool

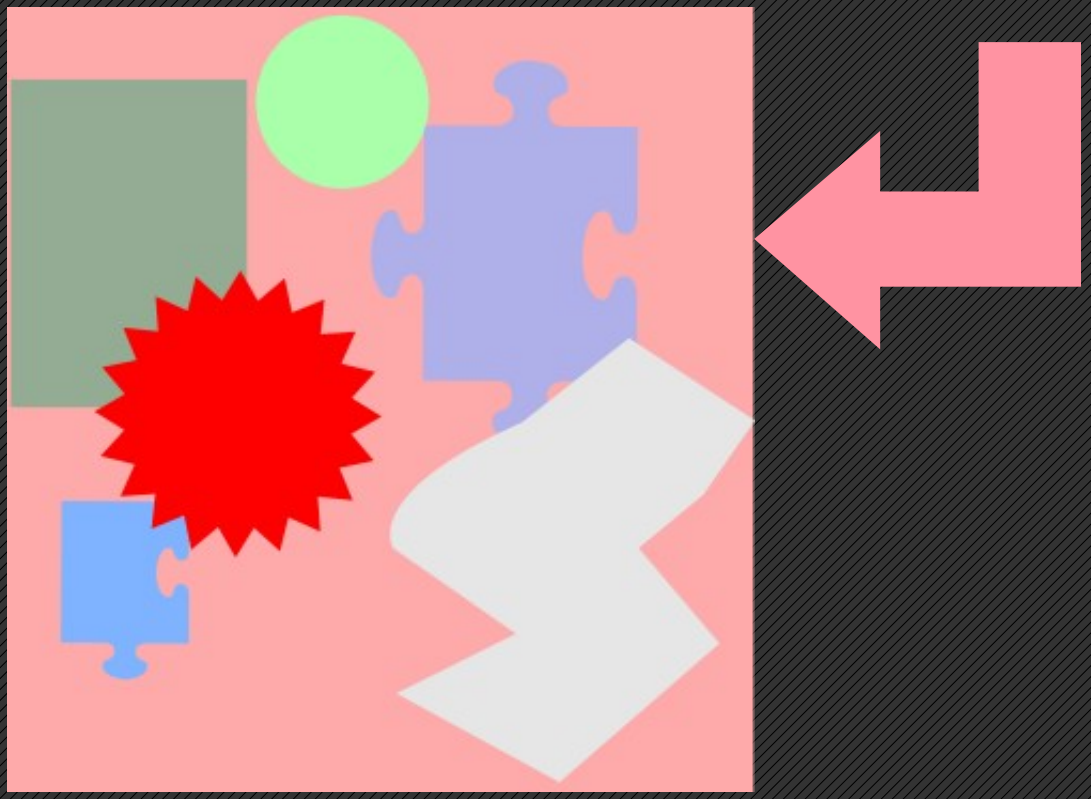


The reality is more like

Integration work

Integration work

Yes, that's the pale red area



Yes, **there's a lot of it**

Integration work

To make it worse, it's usually:

Tightly coupled & brittle
Poorly documented
Poorly tested

Because of integration work,

Cost of change is high

Not all tools are created equal

Solutions?

(in the context of a toolchain)

Gratuitous, out of context, quote:

“Tame the tool sprawl”

(Damon Edwards)

http://www.slideshare.net/dev2ops/with-selfservice-operations-the-cloud-is-just-expensive-hosting-20-a-devops-story

Leverage your existing tools

Incremental changes

Evaluate tool change only if it improves your workflow

workflow

Noun

the sequence of industrial, administrative, or other processes through which a piece of work passes from initiation to completion.

Pick your poison

Lean
Just-in-Time
Kanban
Theory of Constraints

Adding a new tool should be a conscious decision, trying to take into account the Cost/Benefit.

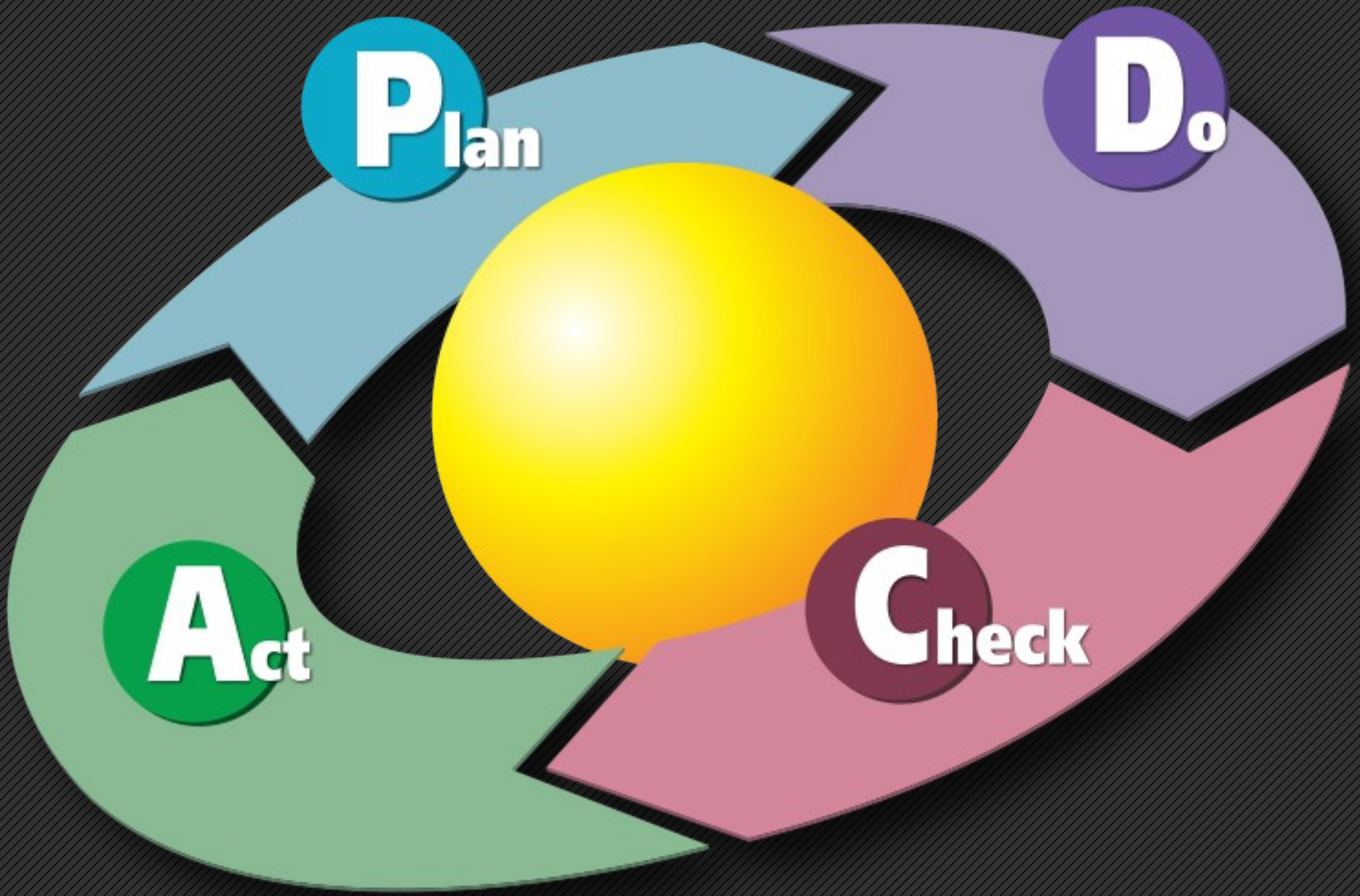
YES

Evaluate new tools

You need to know them, their Pros and Cons before being able to make a conscious decision

BUT

be ready to ditch them if they don't provide the expected benefits.


Plan
Do
Check
Act

In fact,

Test Test Test

If there was one tool, it would be

CI/CD

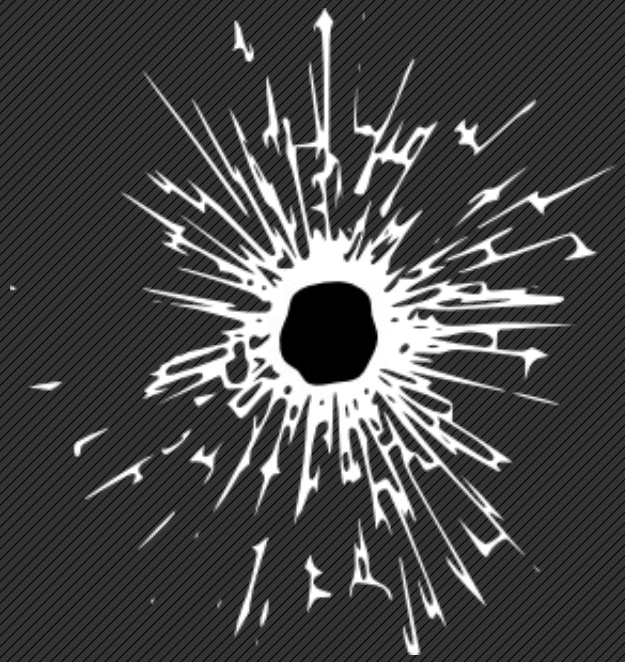
“Everything that's not continuously tested **will** bit rot.”

Create a virtuous circle by building on solid foundations

Payback your technical debt

Enforce a predictable workflow

Fight the “broken window” effect



In conclusion?

It's not (just) the tools
It's mainly the workflow