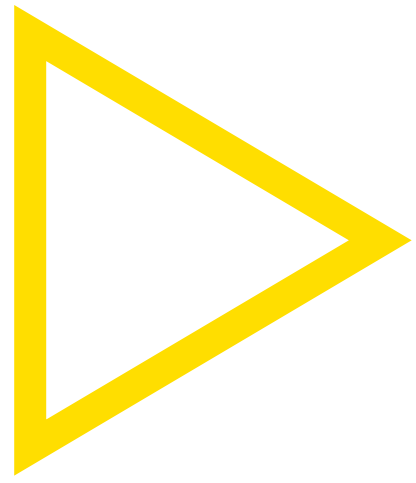


19 essential DevOps ingredients for SAP

PRACTICAL THINGS YOU CAN DO TODAY TO
ACCELERATE YOUR DEVOPS JOURNEY

Resulting^{it}





ACCELERATE YOUR
DEVOPS JOURNEY

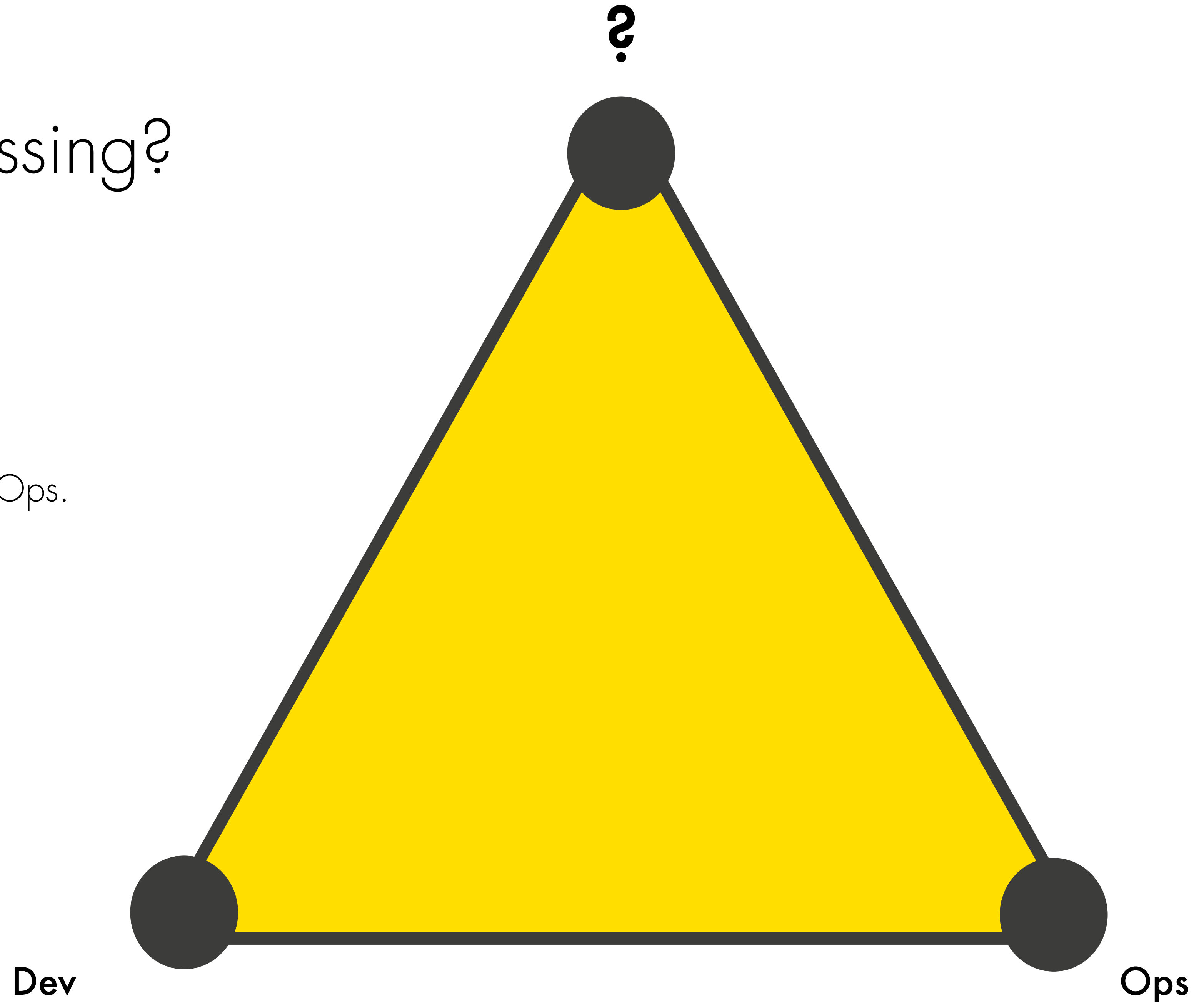
Something missing?

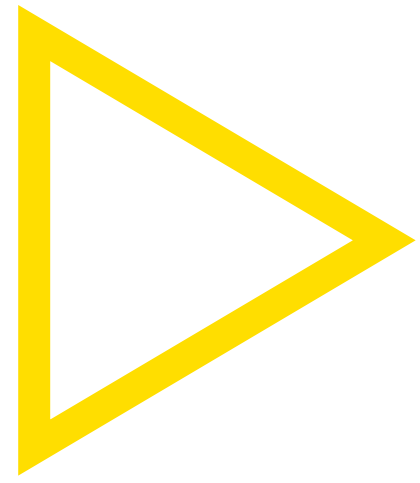
You know about Dev.

You know about Ops.

You might know about DevOps.

But something is missing...

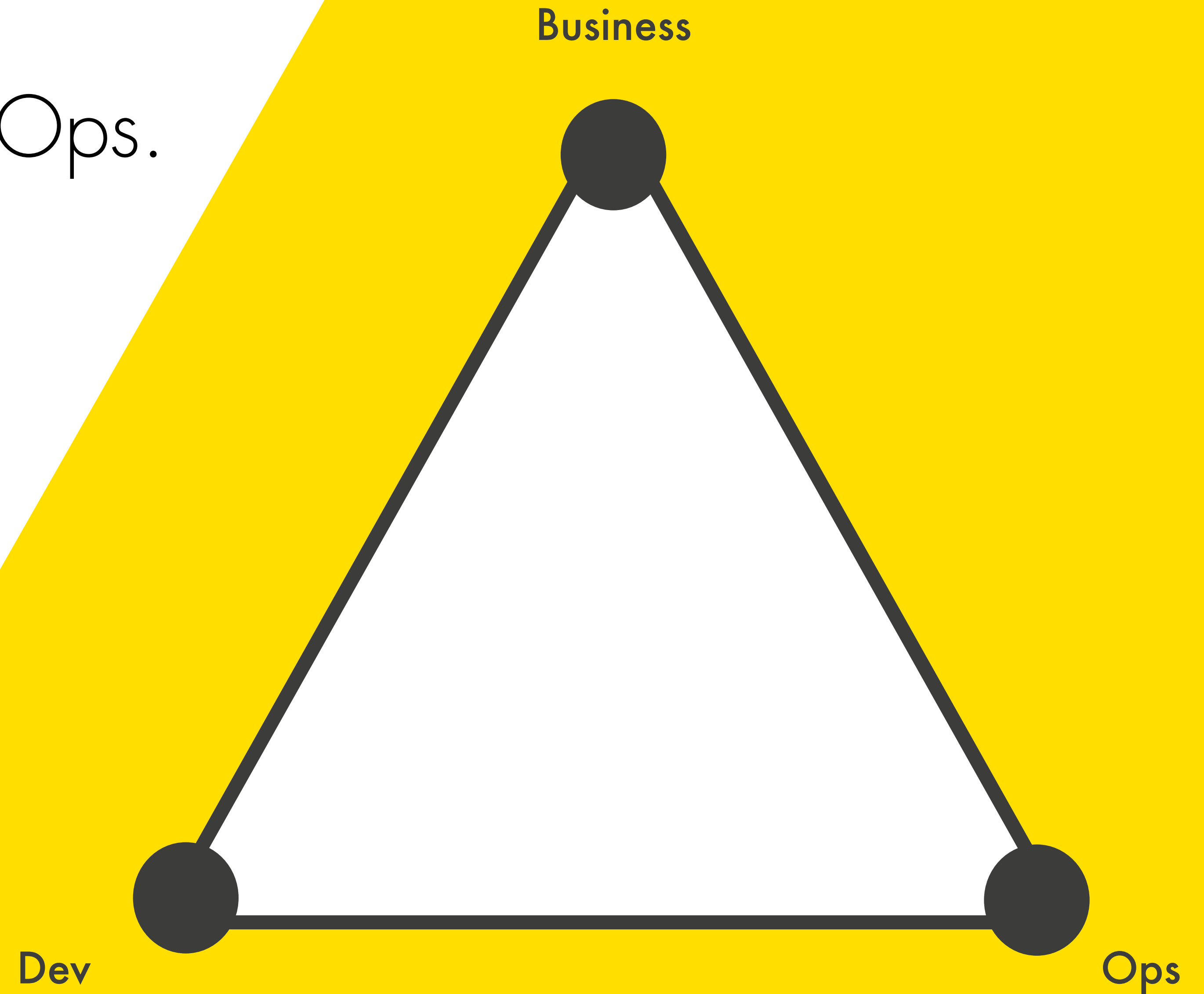


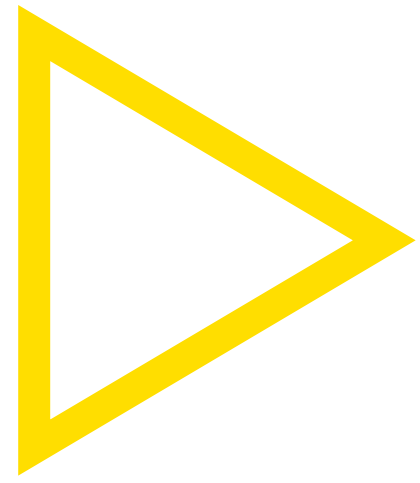


ACCELERATE YOUR
DEVOPS JOURNEY

Introducing BusDevOps.

In any organisation, the Business, Development and Operations often sit as separate entities...

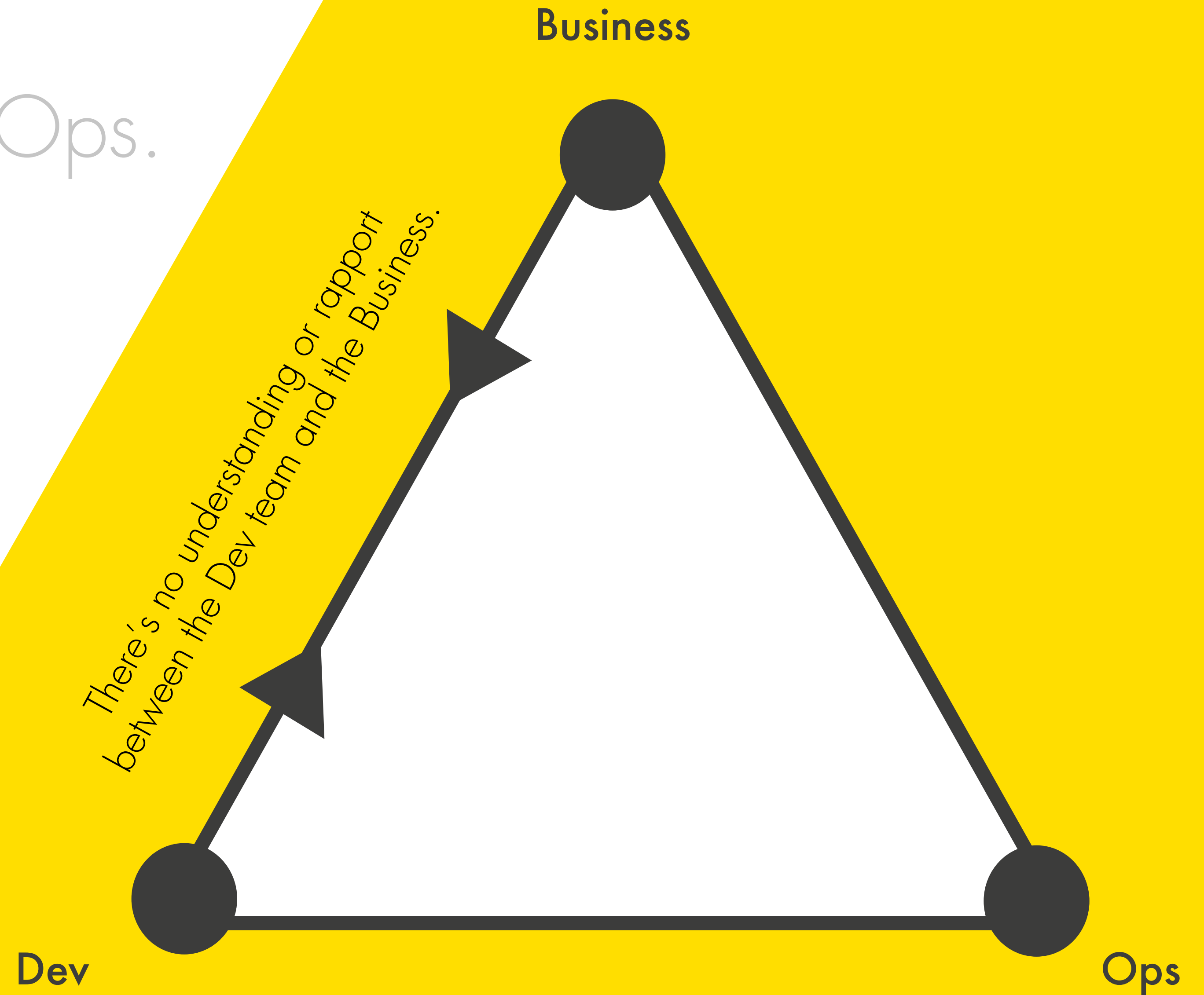


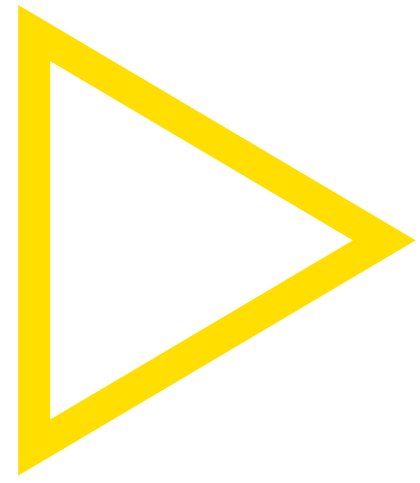


ACCELERATE YOUR DEVOPS JOURNEY

Introducing BusDevOps.

In any organisation, the Business, Development and Operations often sit as separate entities...

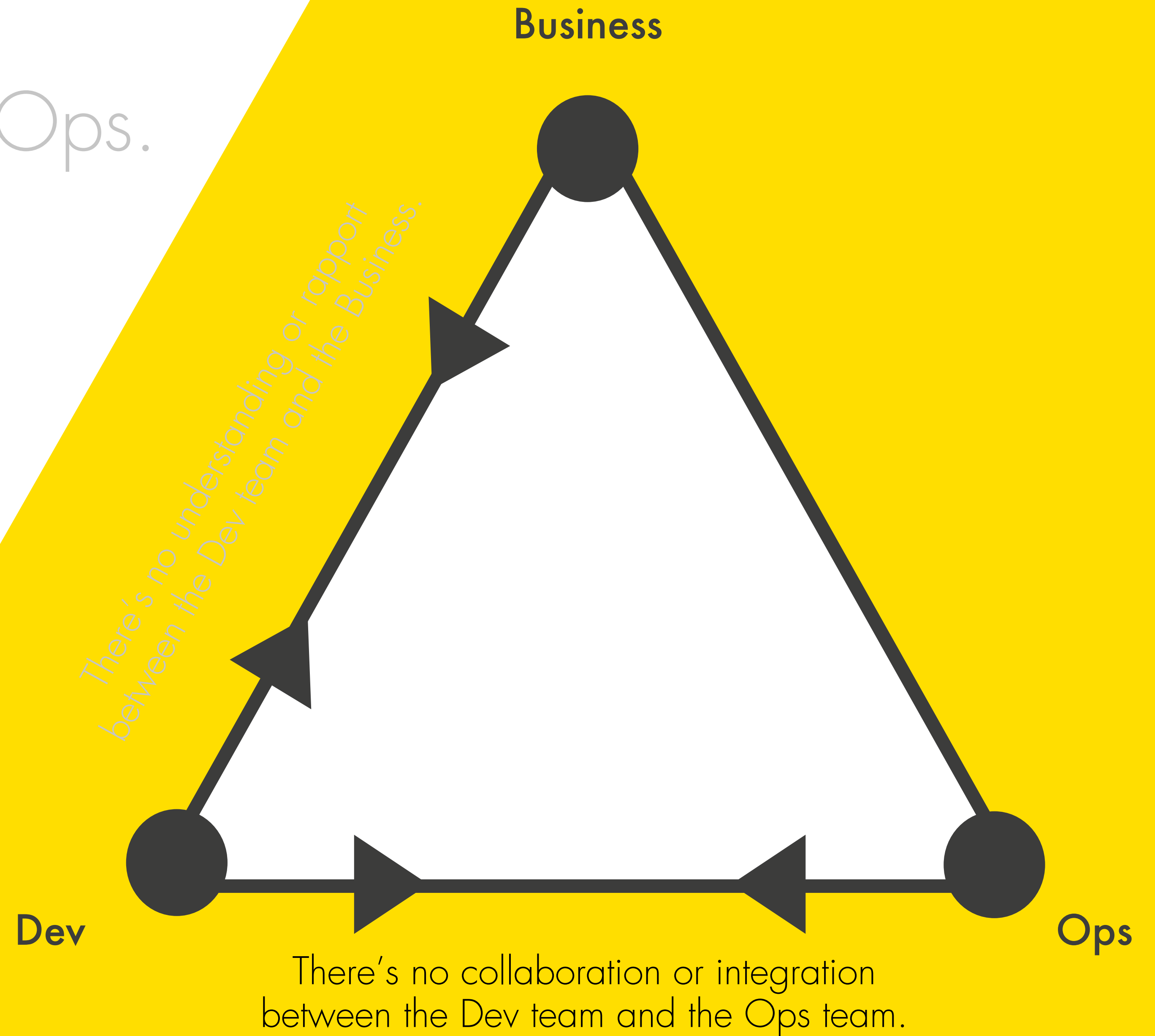




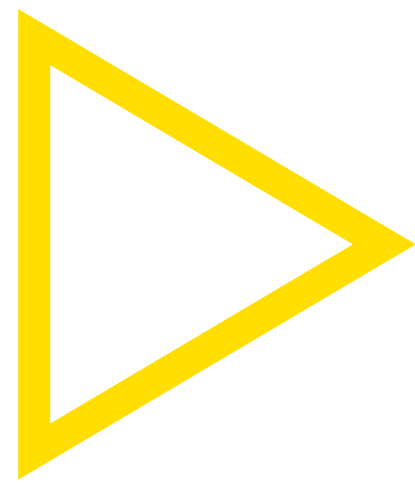
ACCELERATE YOUR DEVOPS JOURNEY

Introducing BusDevOps.

In any organisation, the Business, Development and Operations often sit as separate entities...

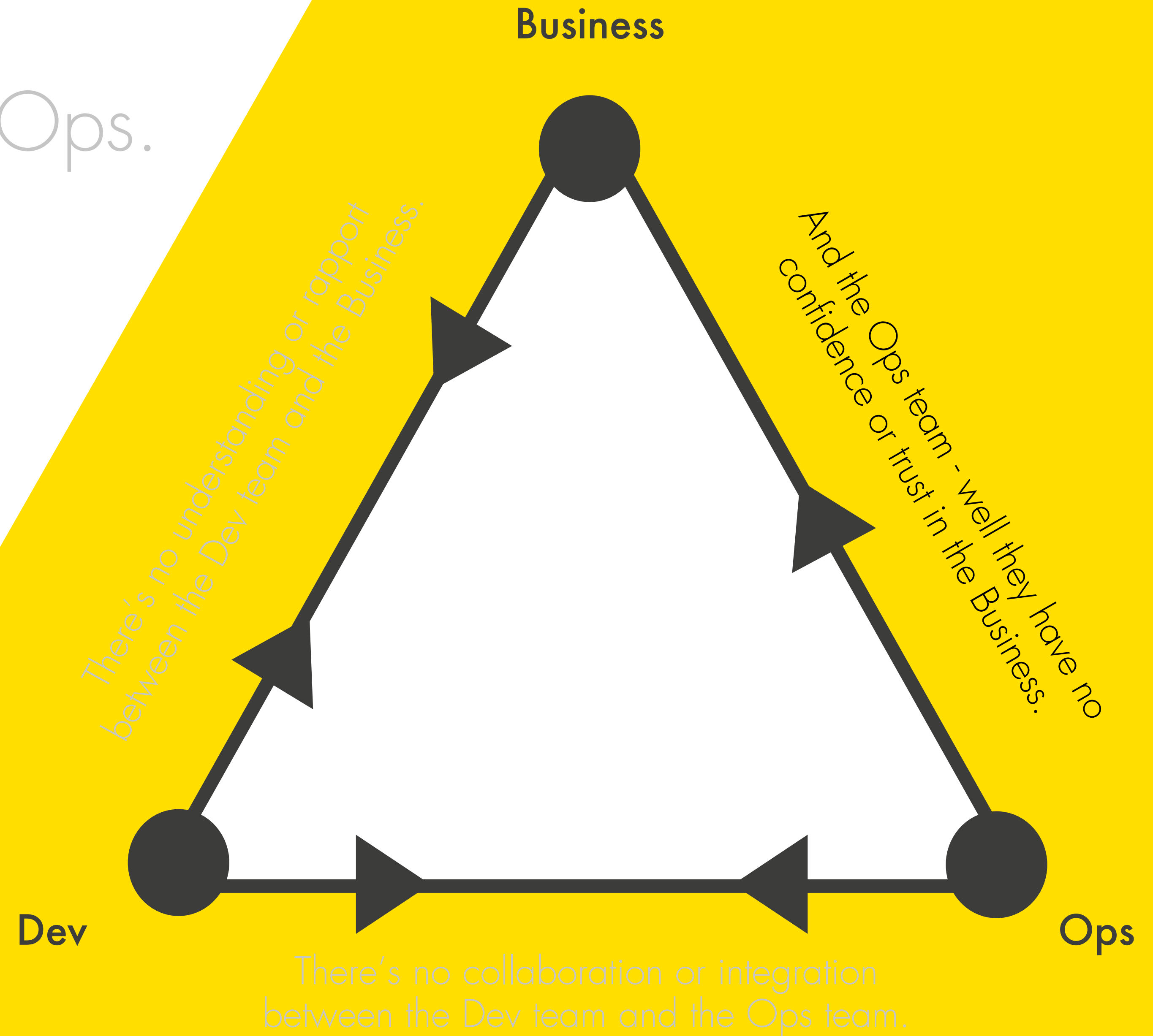


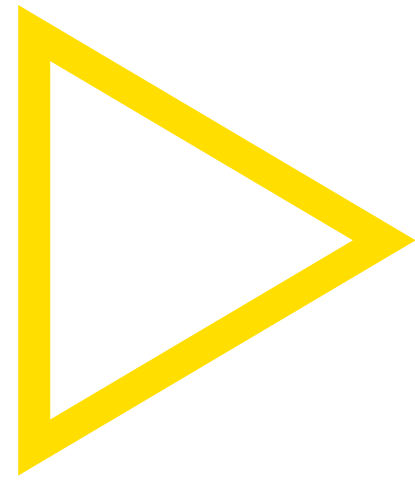
ACCELERATE YOUR DEVOPS JOURNEY



Introducing BusDevOps.

In any organisation, the Business, Development and Operations often sit as separate entities...





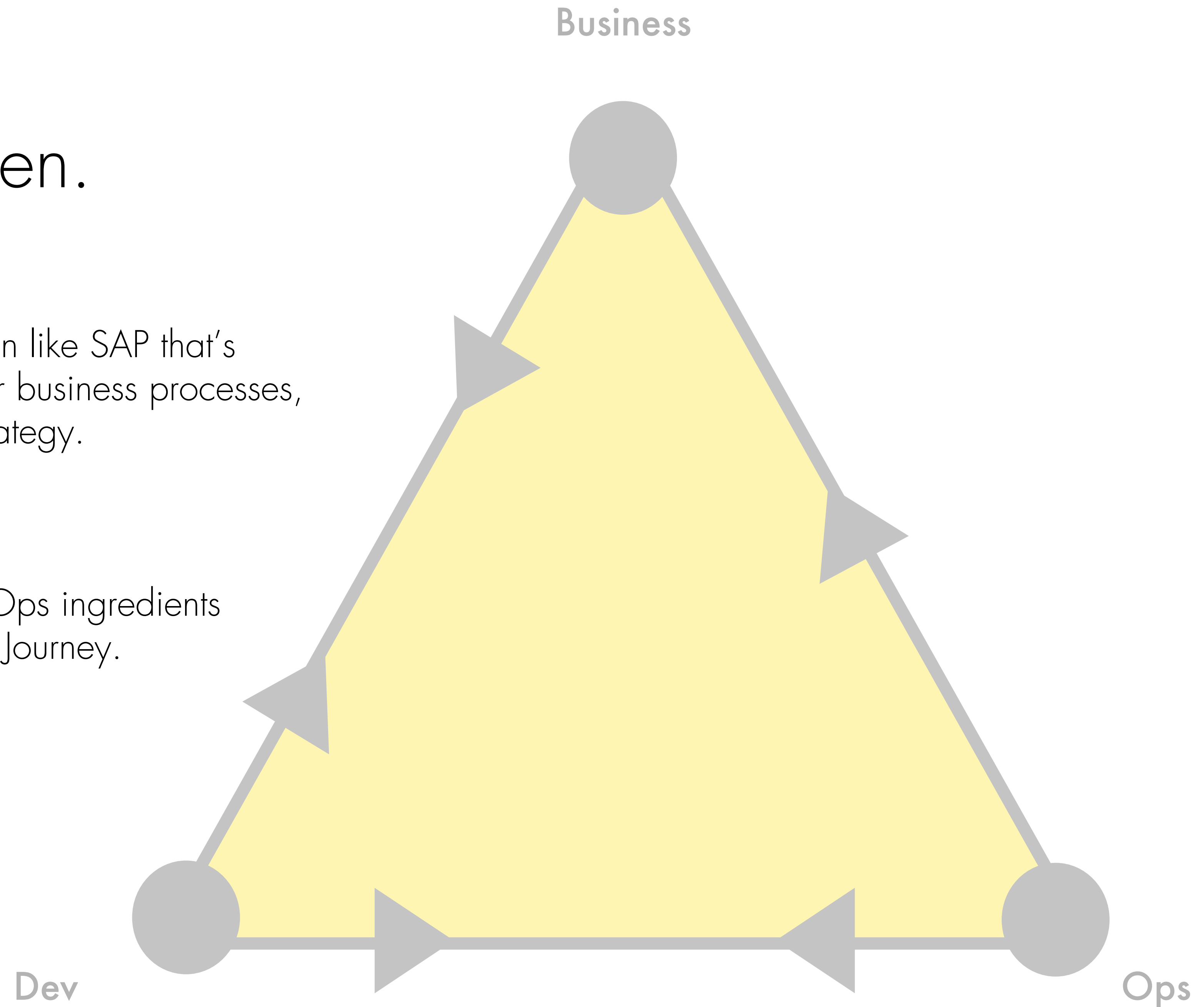
ACCELERATE YOUR DEVOPS JOURNEY

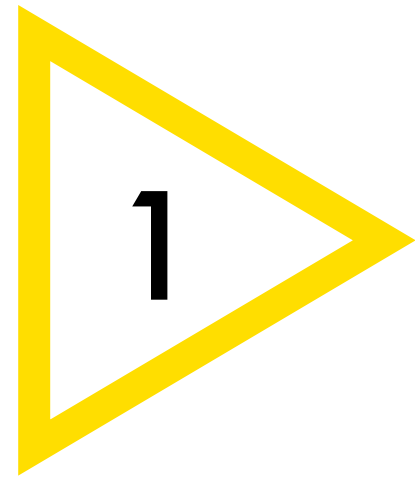
Make it happen.

If you run a complex solution like SAP that's viscerally connected to your business processes, these issues impact your strategy.

So, what can you do?

Here are 19 essential DevOps ingredients to accelerate your DevOps Journey.





UNDERSTANDING & RAPPORT

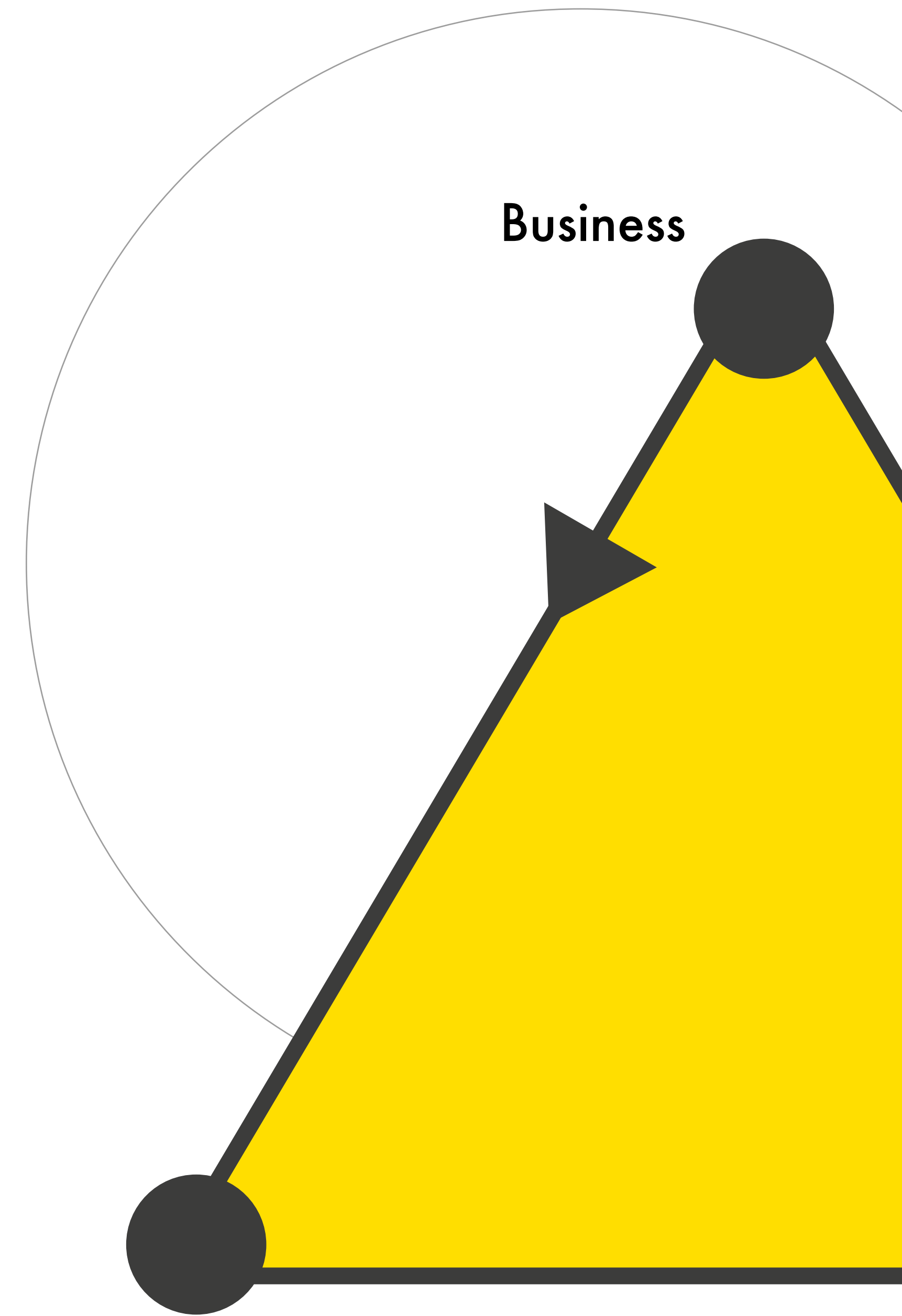
You need a clear, well articulated business vision

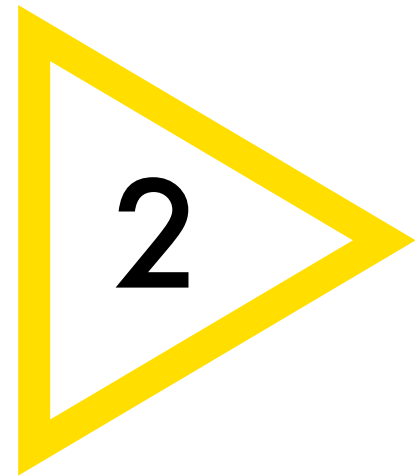
So that Dev know why you're trying to do what you're asking for.

Things get so much easier when Dev have the full context.

Dev

Business



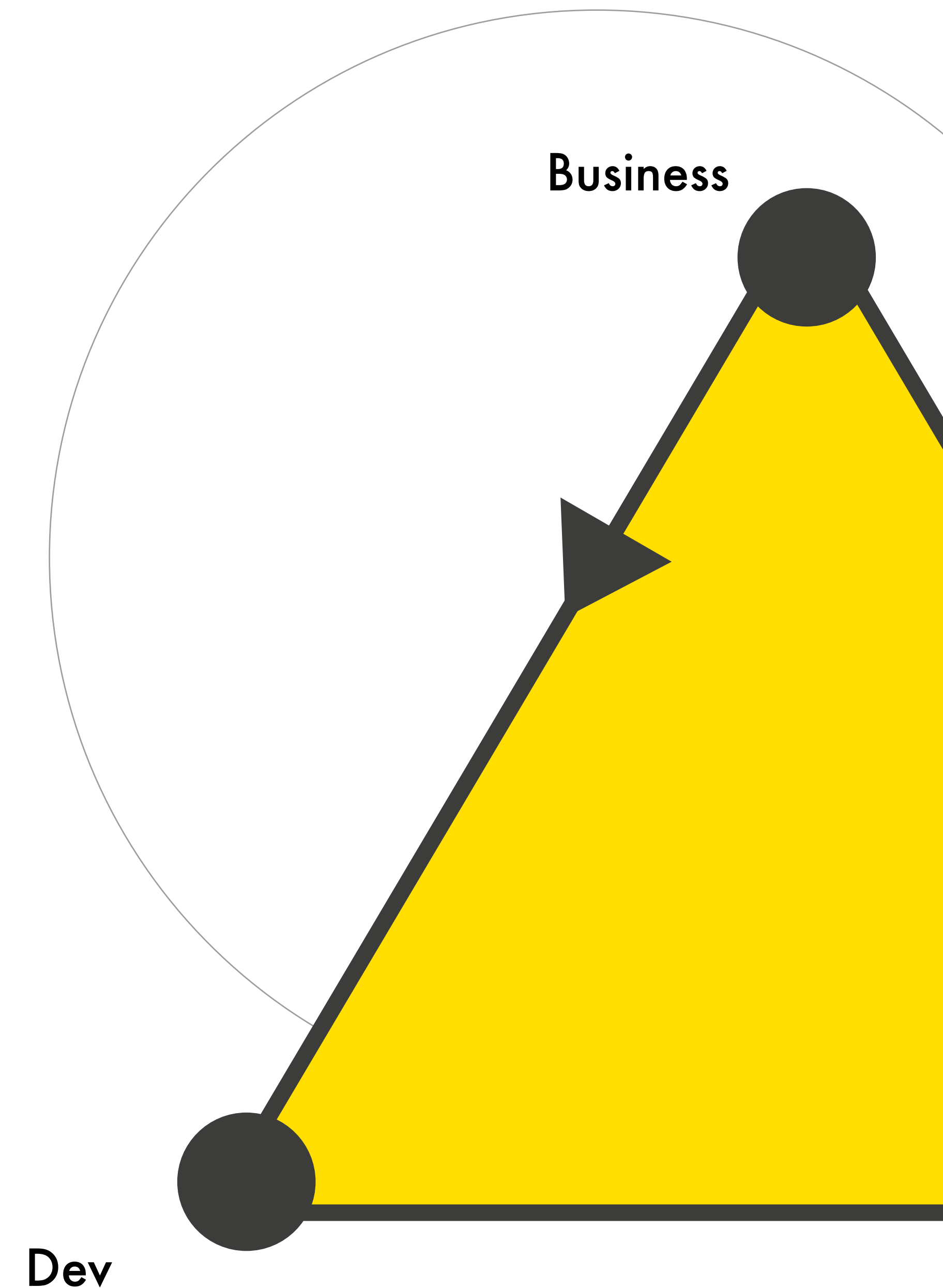


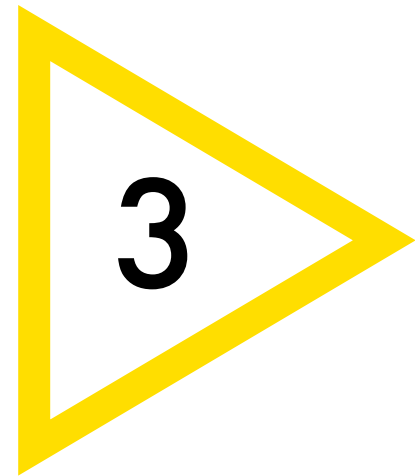
Know how to create a business roadmap for IT

So that Dev knows what's coming next and how it all fits together.

It's much easier creating today's stuff when you know what might be bolted on later.

UNDERSTANDING
& RAPPORT



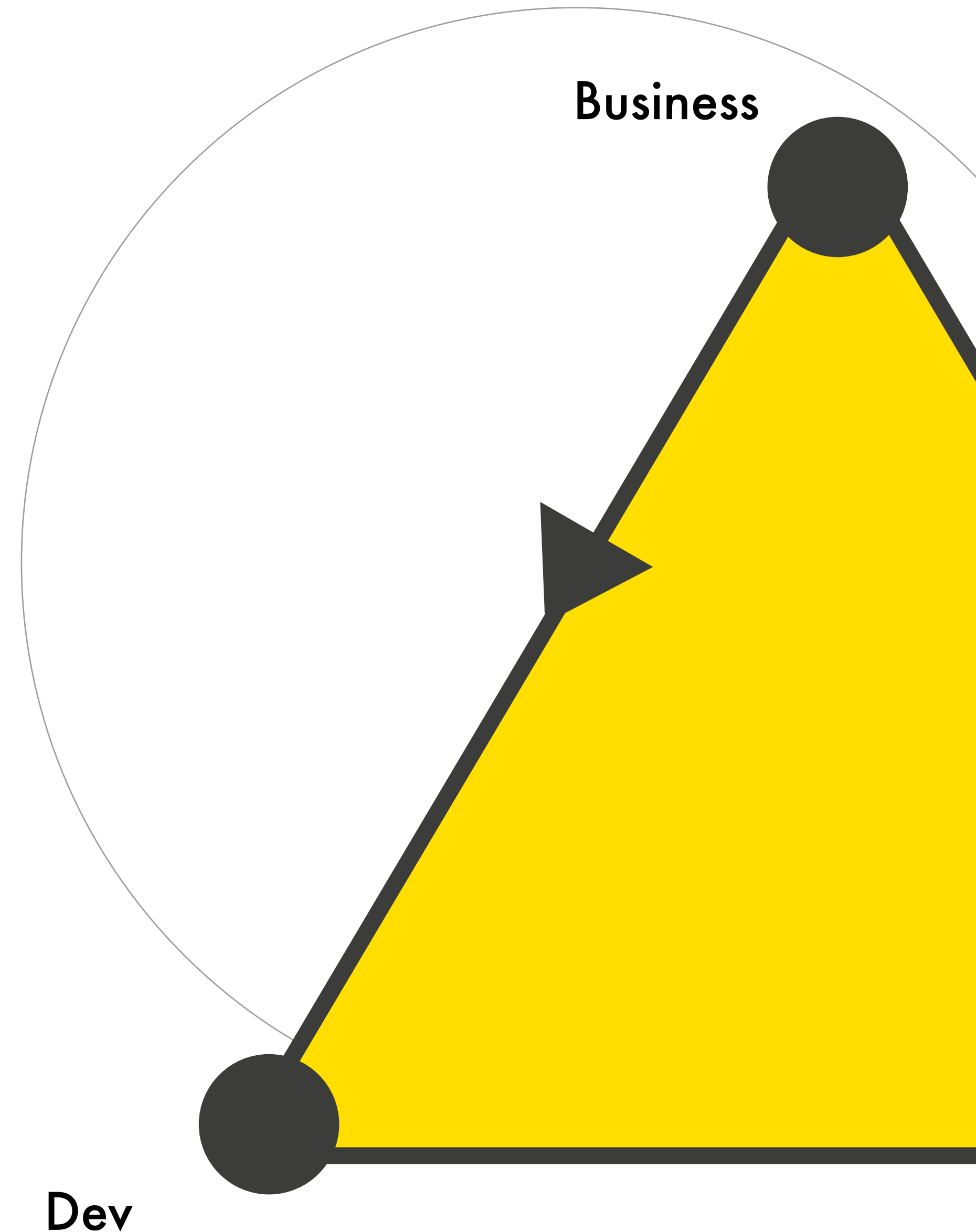


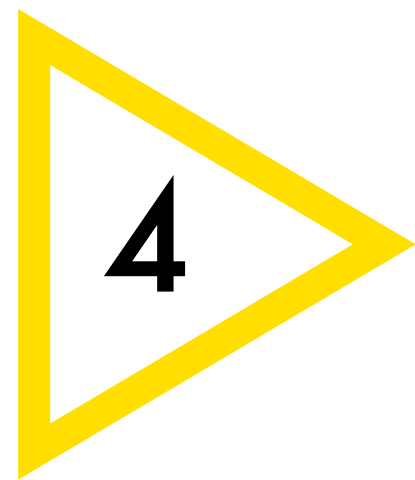
Don't be too prescriptive on everything

Give Dev the ability to contribute the 'how' to the Business 'why'.

Free-range, corn fed IT tastes so much better than the battery variety.

UNDERSTANDING
& RAPPORT





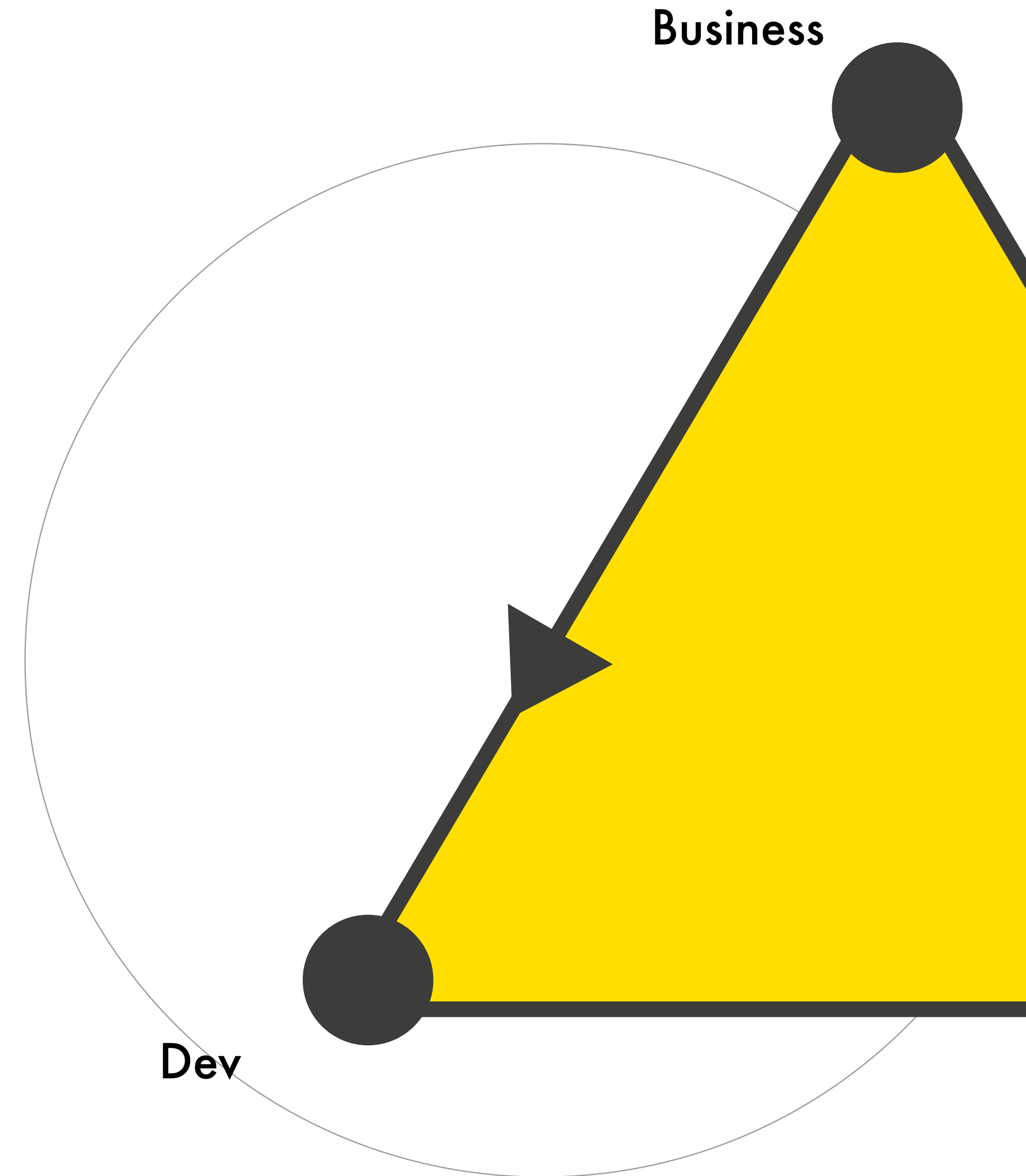
Stop changing your mind so much

This will prevent Dev having to re-do code again and again.

If you don't you'll, wind up with code that's hard to maintain – which will make future releases slower and less reliable.

Some people call this 'technical debt', but we call it wasting money and time.

That's 'time' as in time-to-market. You know, the thing that gives your business its competitive edge.



5

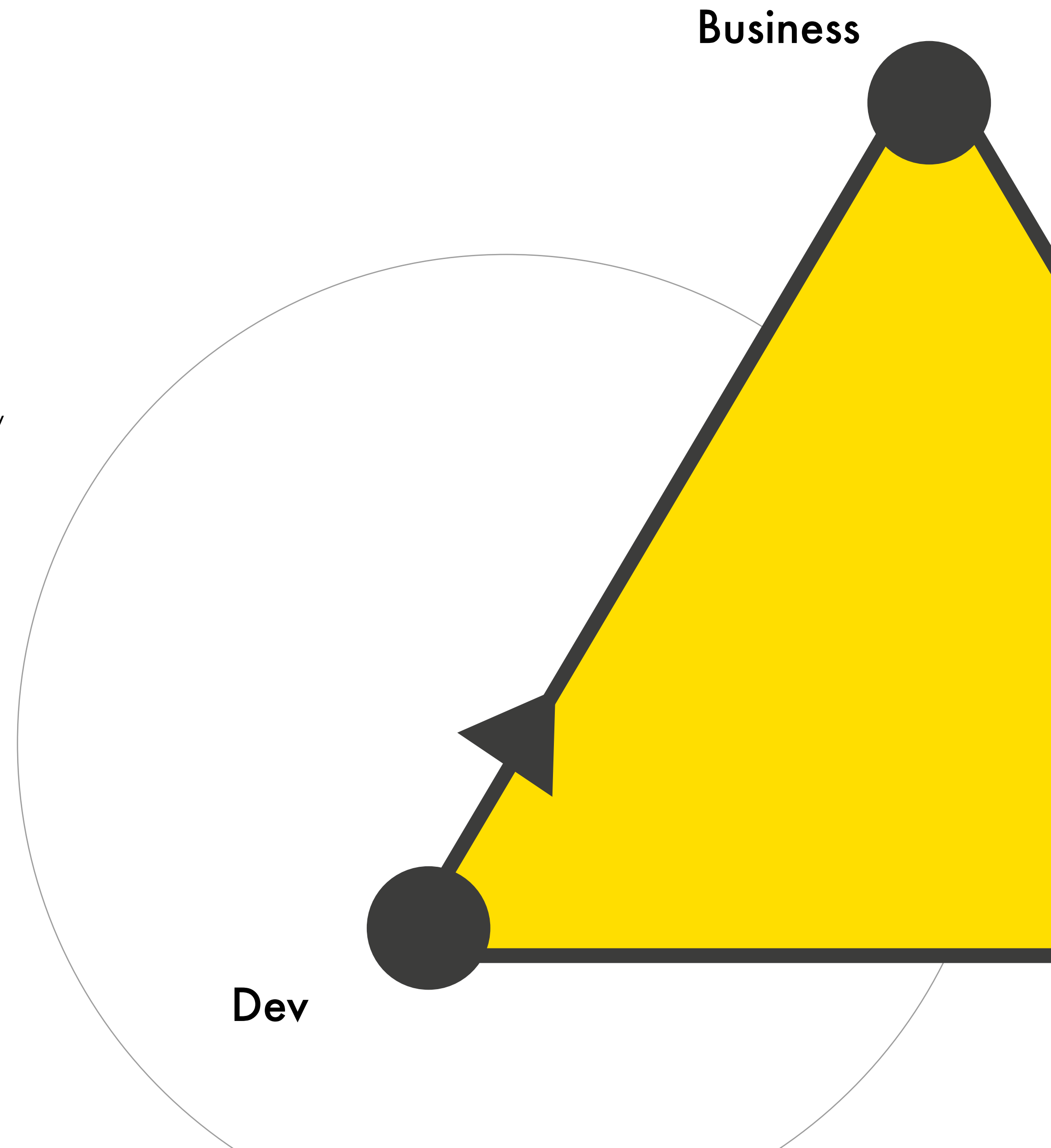
Work out how to ask the right questions

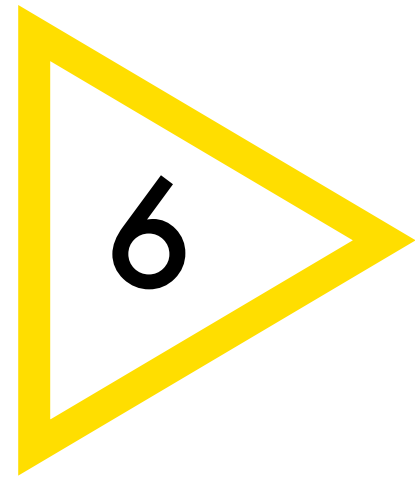
To elicit what the Business 'might' be thinking.

Sometimes asking the same thing in a different way and really listening can completely change the Dev approach to a Business problem.

Listening is a breakthrough cultural change - you have two ears and one mouth for a reason.

UNDERSTANDING
& RAPPORT





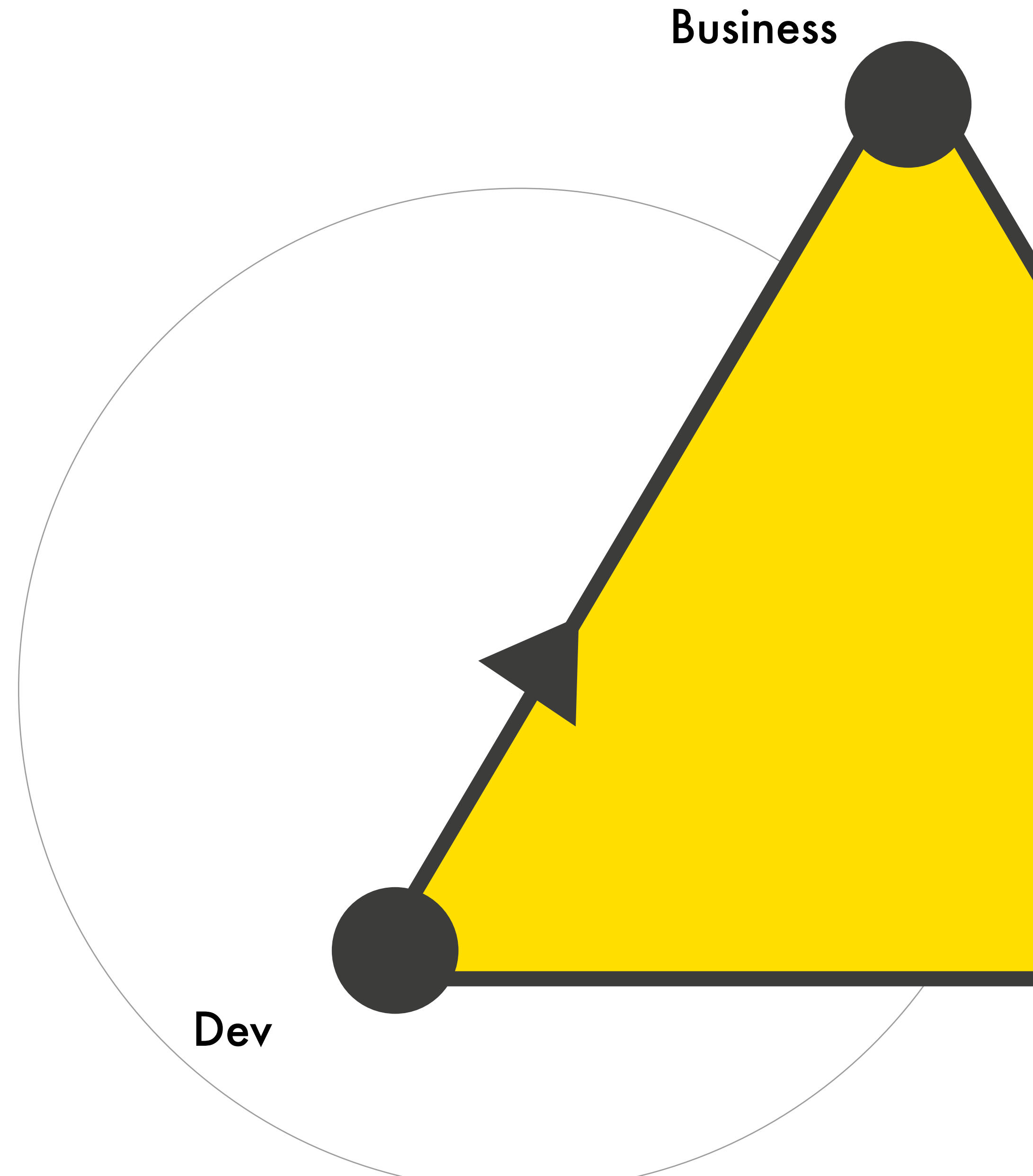
Use simple, non-technical language where possible

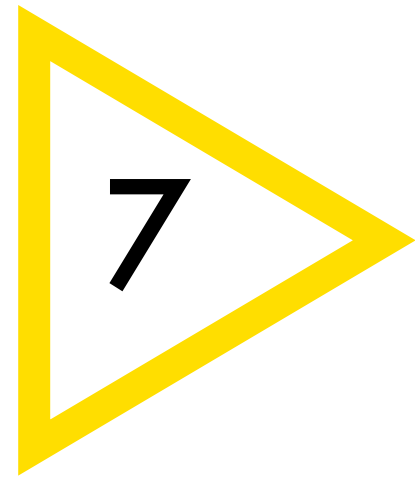
Technical jargon really alienates Business people – especially if it sounds like it's being used to show off.

If you can use simple language, do it.

Pretend you're explaining things to your grandmother.

UNDERSTANDING
& RAPPORT

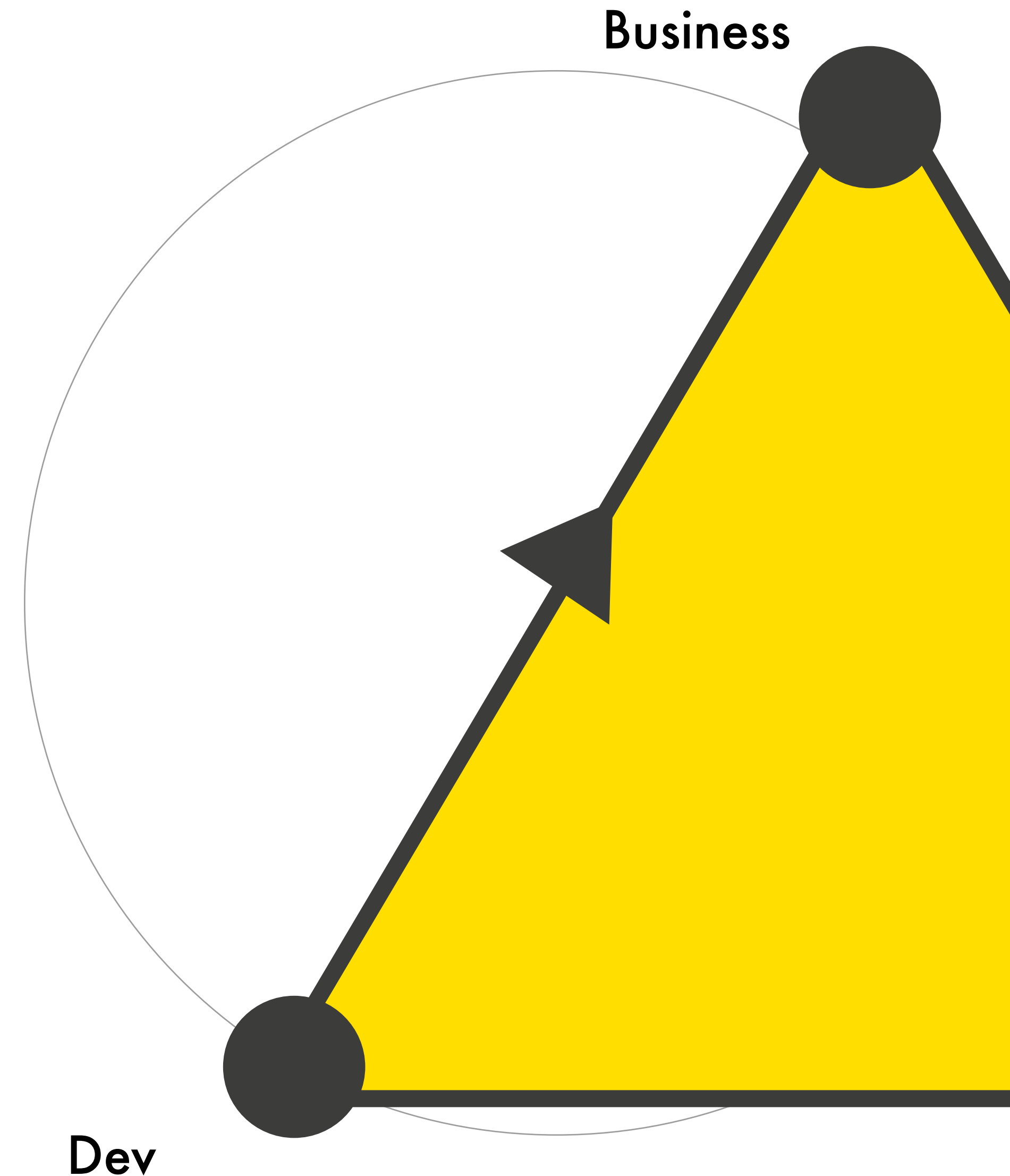


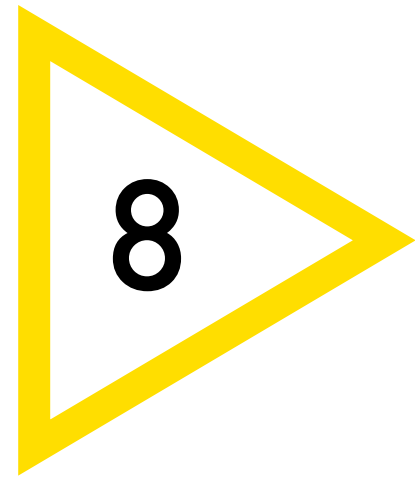


View your technology as the 'possible' and not fact

So that competitive advantage isn't constrained by out-of-the-box, vanilla functionality. If you use an ERP system, it's easy to say "but SAP doesn't work that way".

Every time you hear this, part of your competitive edge over every competitor running SAP dies.





Stimulate innovation and improvement with new technology

Inform the innovation roadmap with the art-of-the-possible.

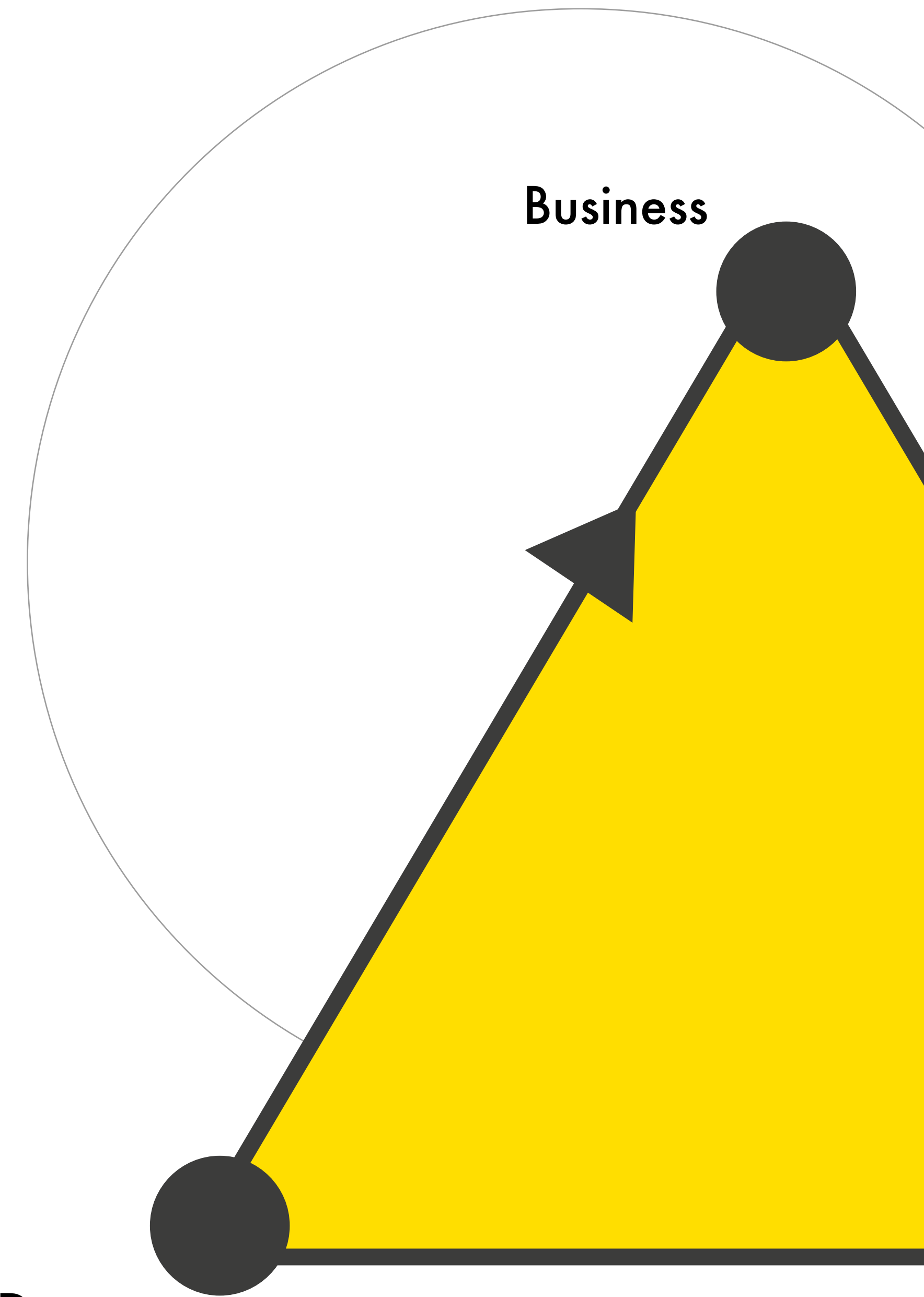
Your technophile Dev guys are much more up to speed on the latest tech than the Business – but they might not know how to apply what they know. So join the dots.

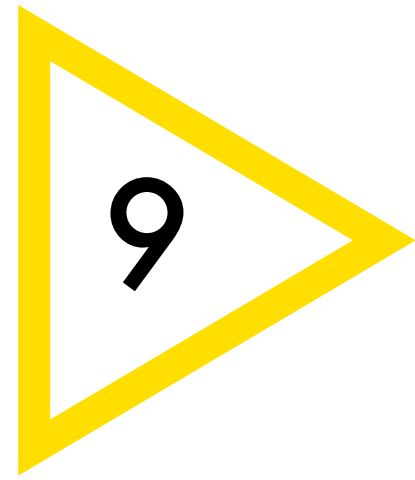
There's so much new, cool, exciting stuff out there.

Just because it's not on your approved architecture list doesn't mean you have to pretend it doesn't exist.

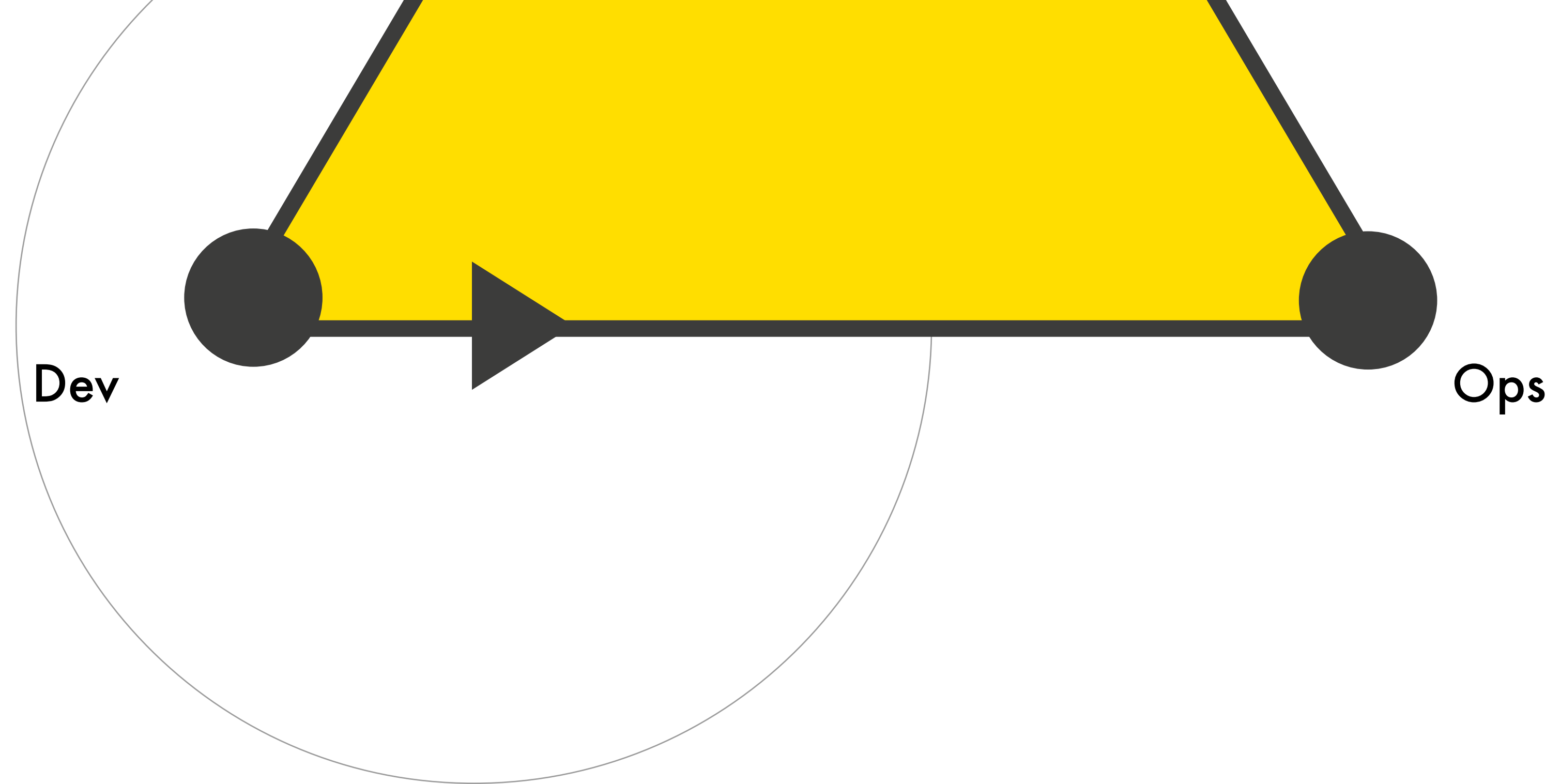
Dev

Business





COLLABORATION & INTEGRATION

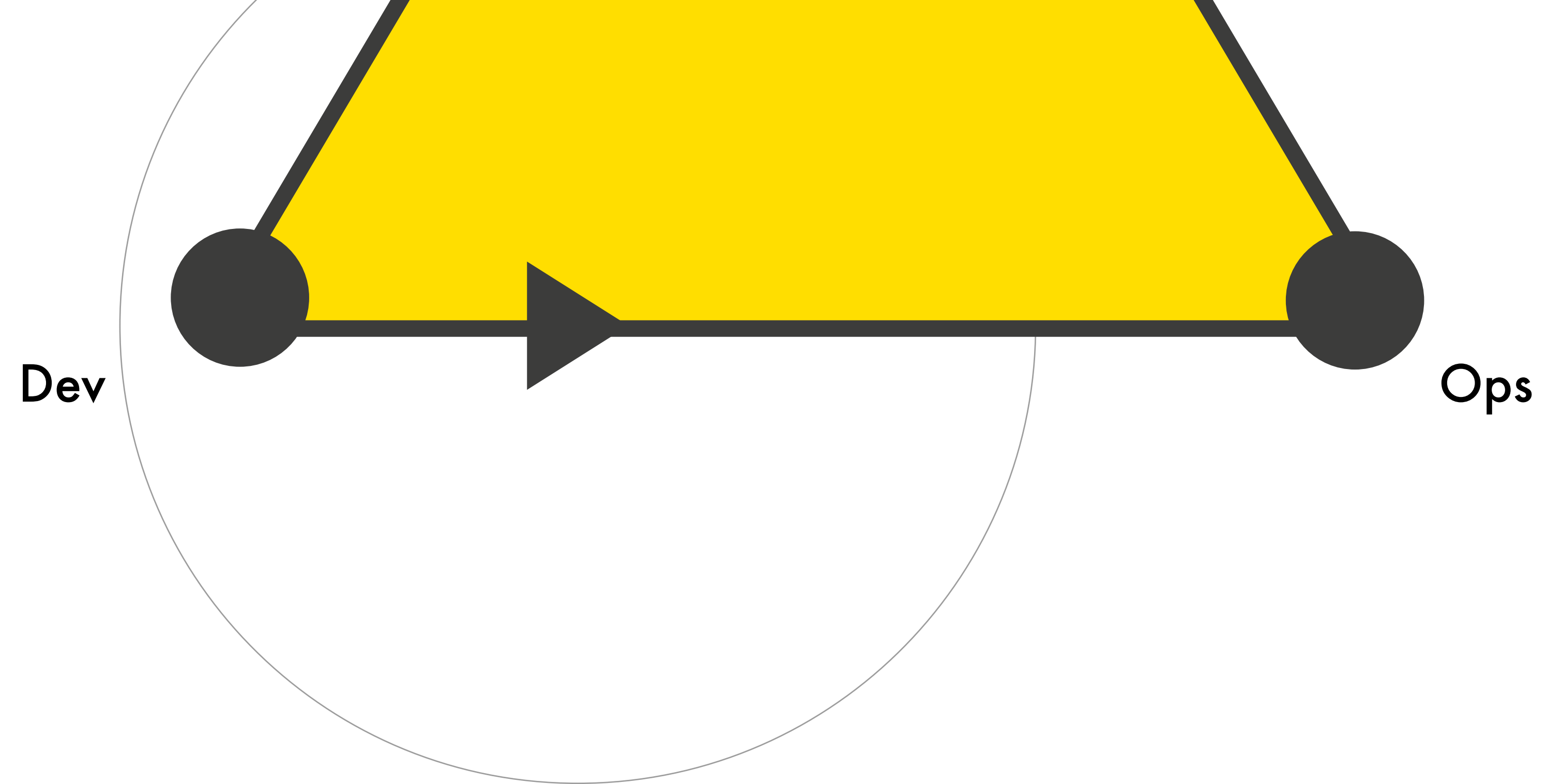


Forget there was ever a wall
between Dev & Ops

This way, Dev doesn't have something to throw stuff over.
Make 'wall' a banned word with a hefty fine (like doughnuts
or pizza...).

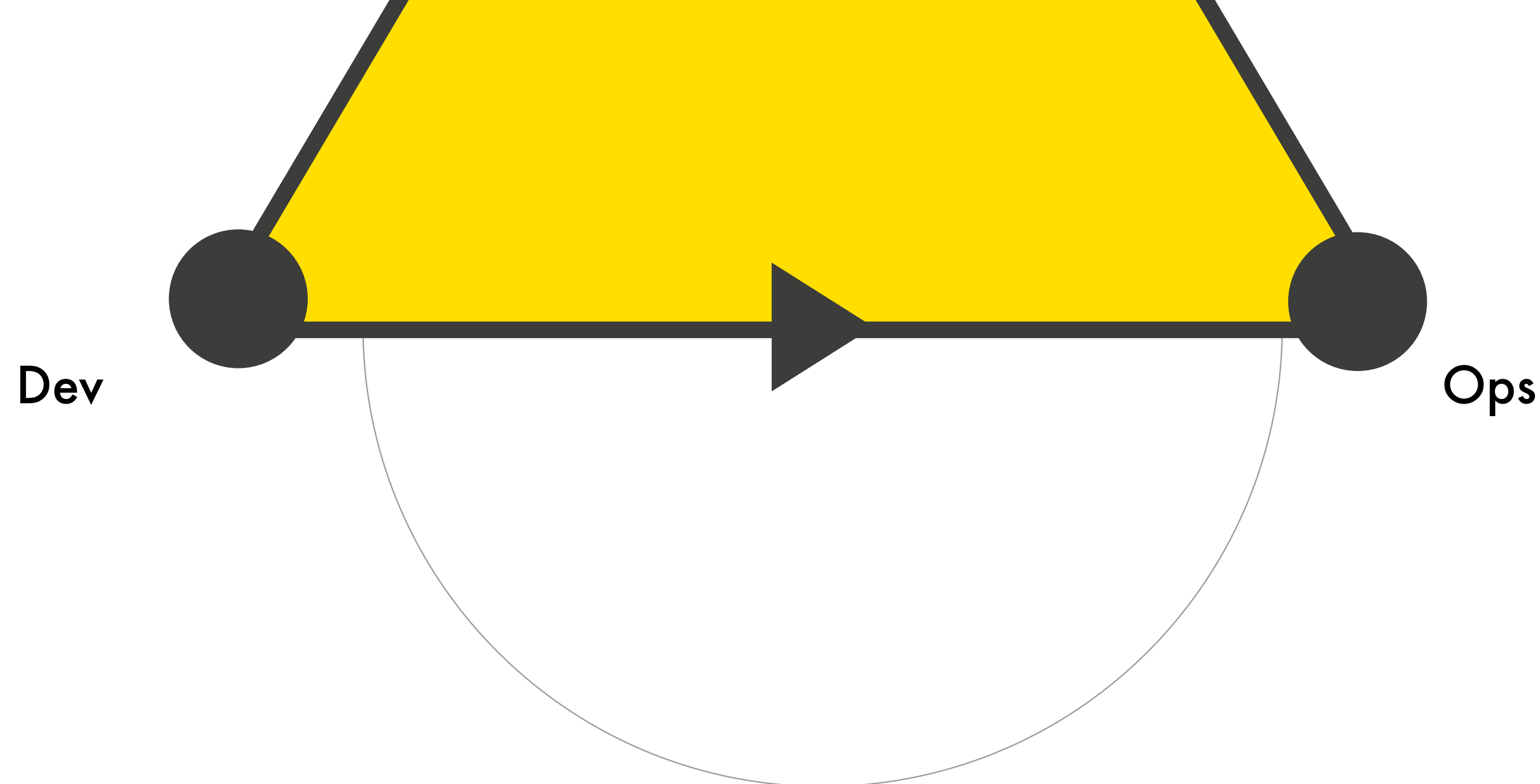
10

COLLABORATION & INTEGRATION



Take responsibility for performance,
availability and recoverability

So that Ops don't have to hope and guess what will
happen when things hit production. Or worse still,
when things go wrong in production.

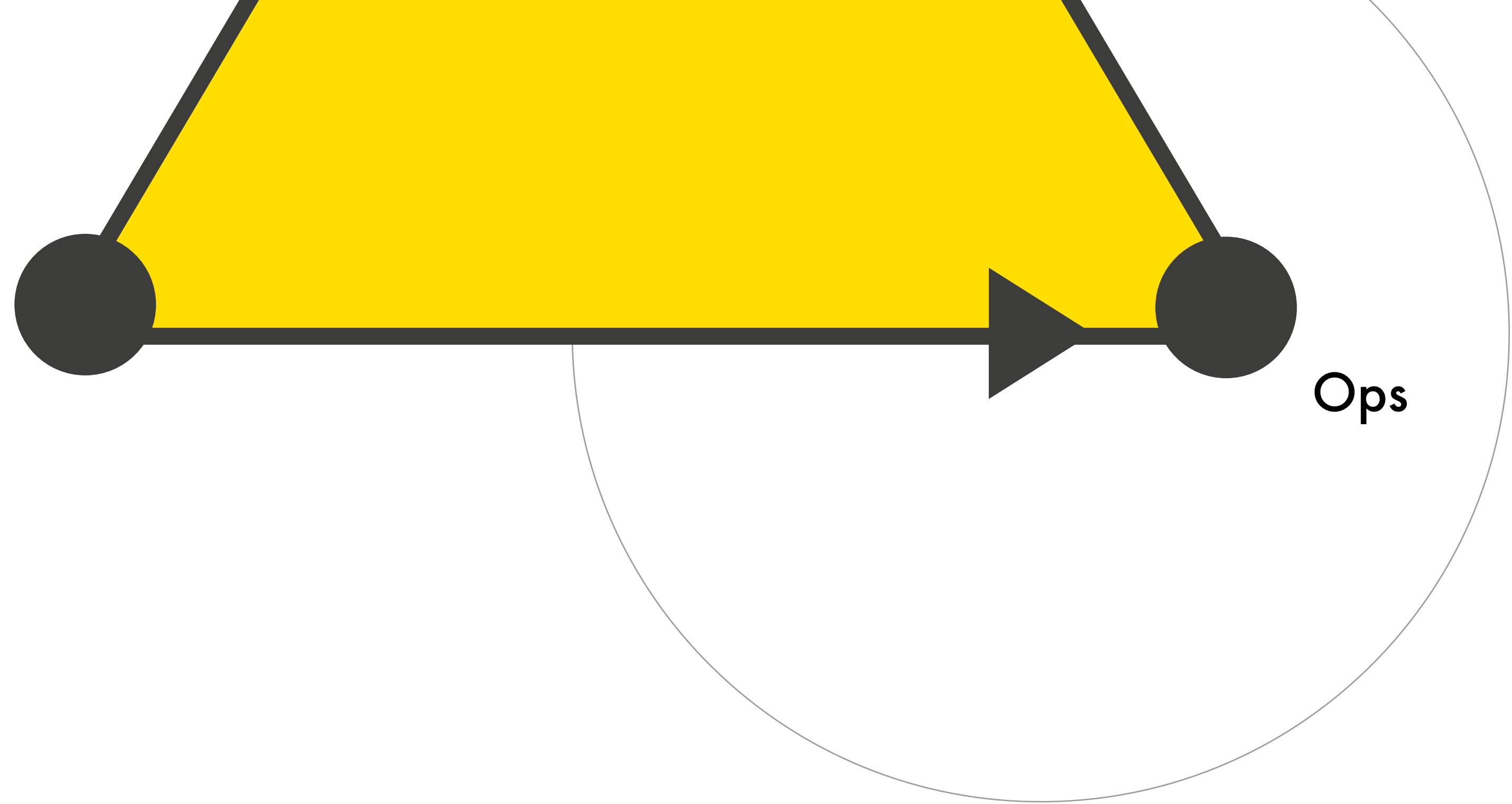


Assume Ops serve the Business and not Dev

Pass Dev output through Ops to the Business and drive trust in IT on behalf of Ops.

Pretend Dev is Obi Wan and Ops is Luke Skywalker. To the audience, Luke is the hero, but he couldn't do it without the guidance of Obi Wan.

Dev



Ops

Lead Business interaction but involve Ops

Ops can only gain confidence and build trust by being present in BusDev conversations.

Use the force Luke.

Dev

Ops

Forget there was ever a
wall between Ops & Dev

So that Ops can't blame Dev for throwing things over it.

And there isn't a wall. Seriously, forget it.

Sorry. This one is included twice, so technically there are
18 things you can do. But it is REALLY important.

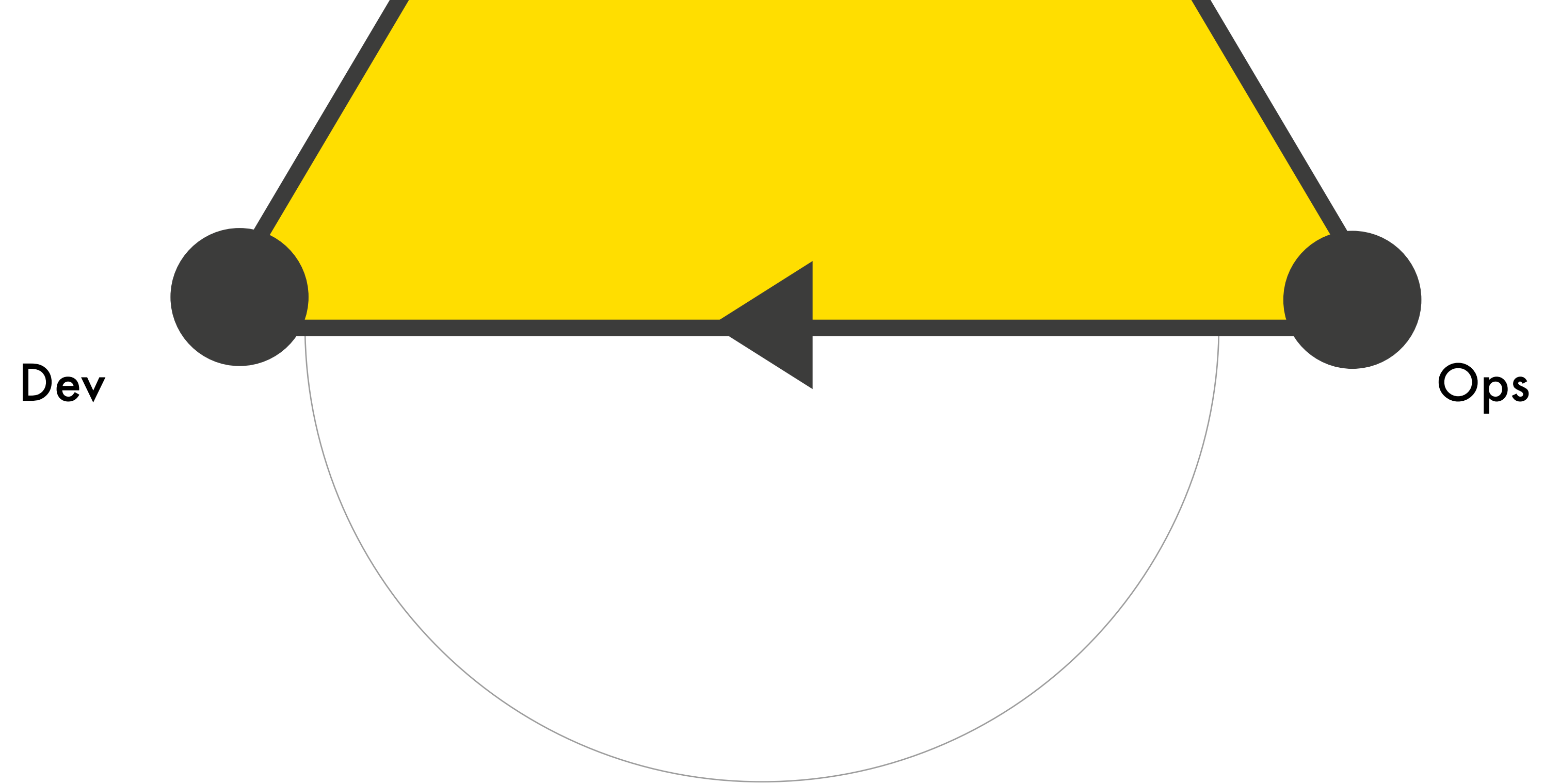
Dev

Ops

Create a slick infrastructure during build – not at go-live

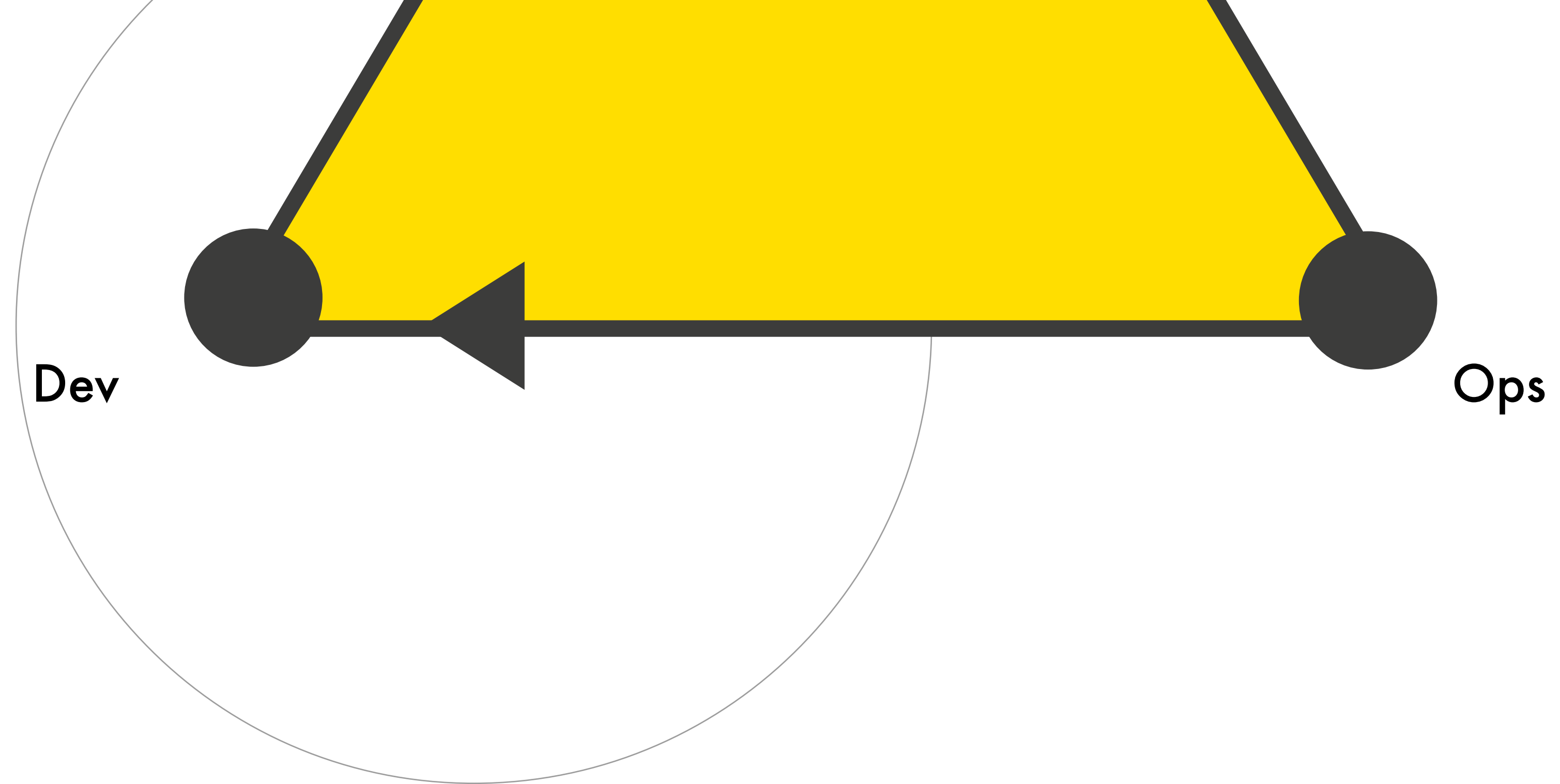
Do this to improve Dev cadence and enable a fast, smooth build. Sure, Prod infrastructure is important – but Dev and Test needs to be just as well thought out.

And your Dev > Test > Prod workflow will be there forever – so get it right at the start.



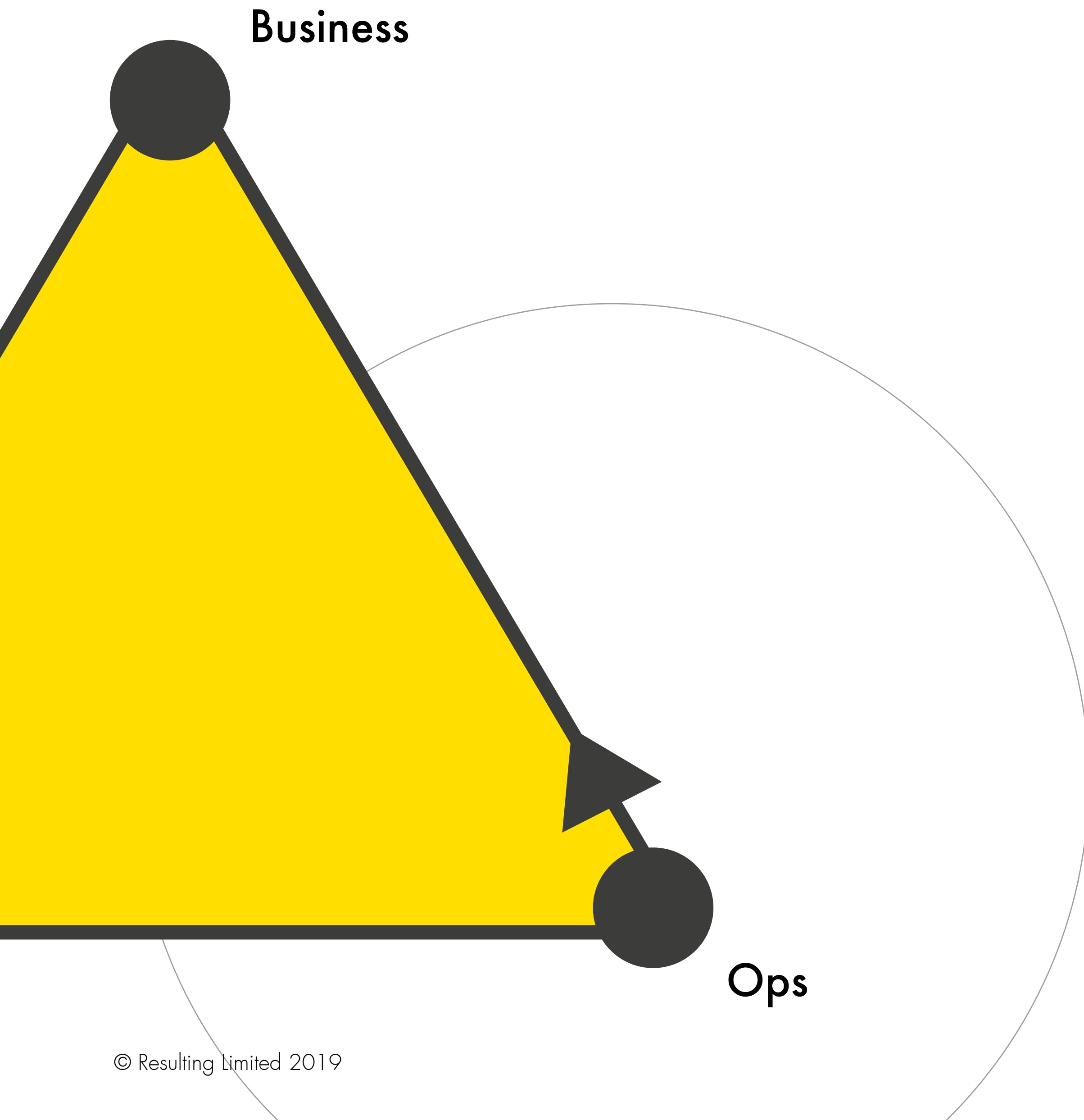
Learn to create fast, production-like environments

This leads to faster, better testing – and therefore faster, better releases. And, it also means that the Business gains confidence in early testing of Dev builds - which helps everybody.



Pull builds towards the Business rather than pushing them back to Dev

Pushing back on Dev slows momentum and doesn't position Ops to be the trustworthy party. Ops concerns over LPARs and Blades don't correlate directly to business strategy – but on-time delivery does. So focus on what matters most to the Bus guys.

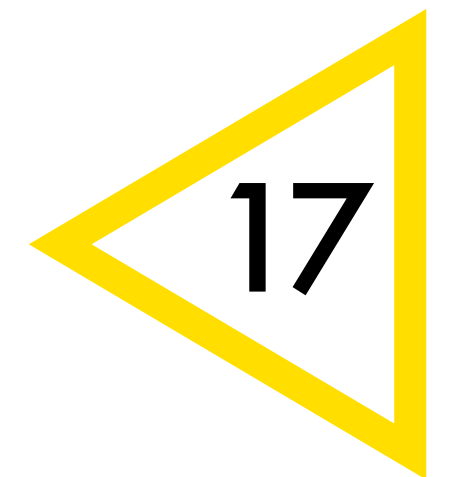


Re-think Non-Functional Requirements

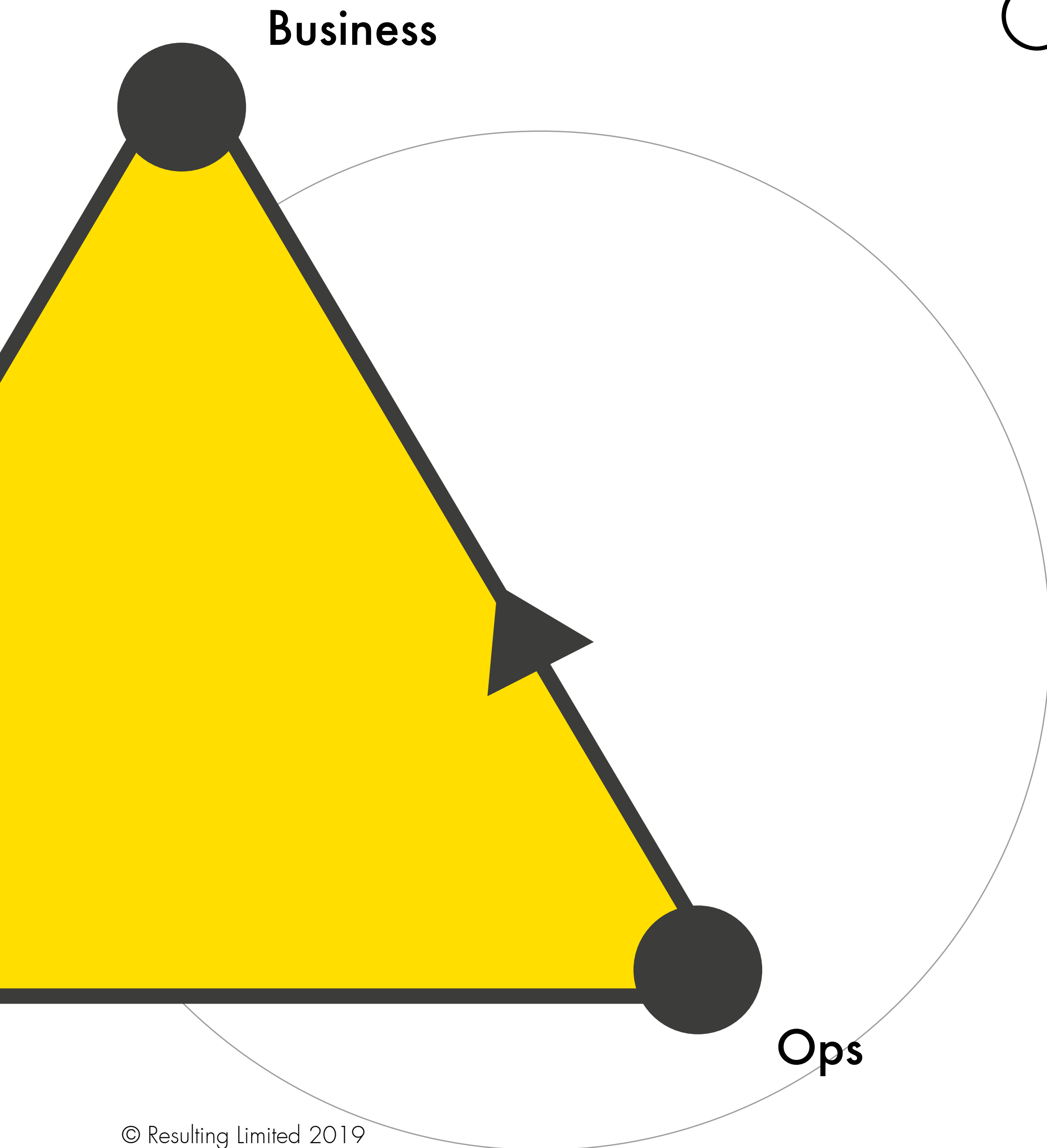
Because functional Business people don't understand or value them.

Instead, find out what Business people want and deliver it.

In practice – not on paper.



CONFIDENCE
& TRUST

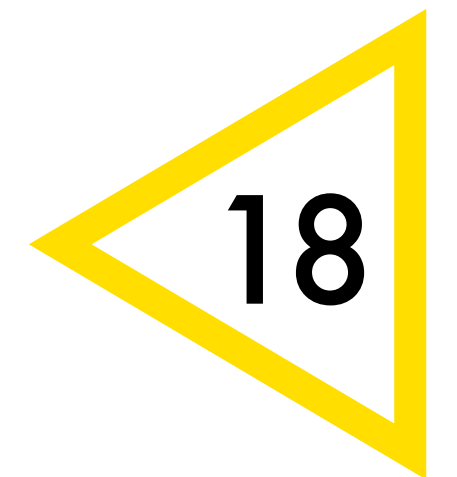


Check-back from time-to-time.
Like a good waiter.

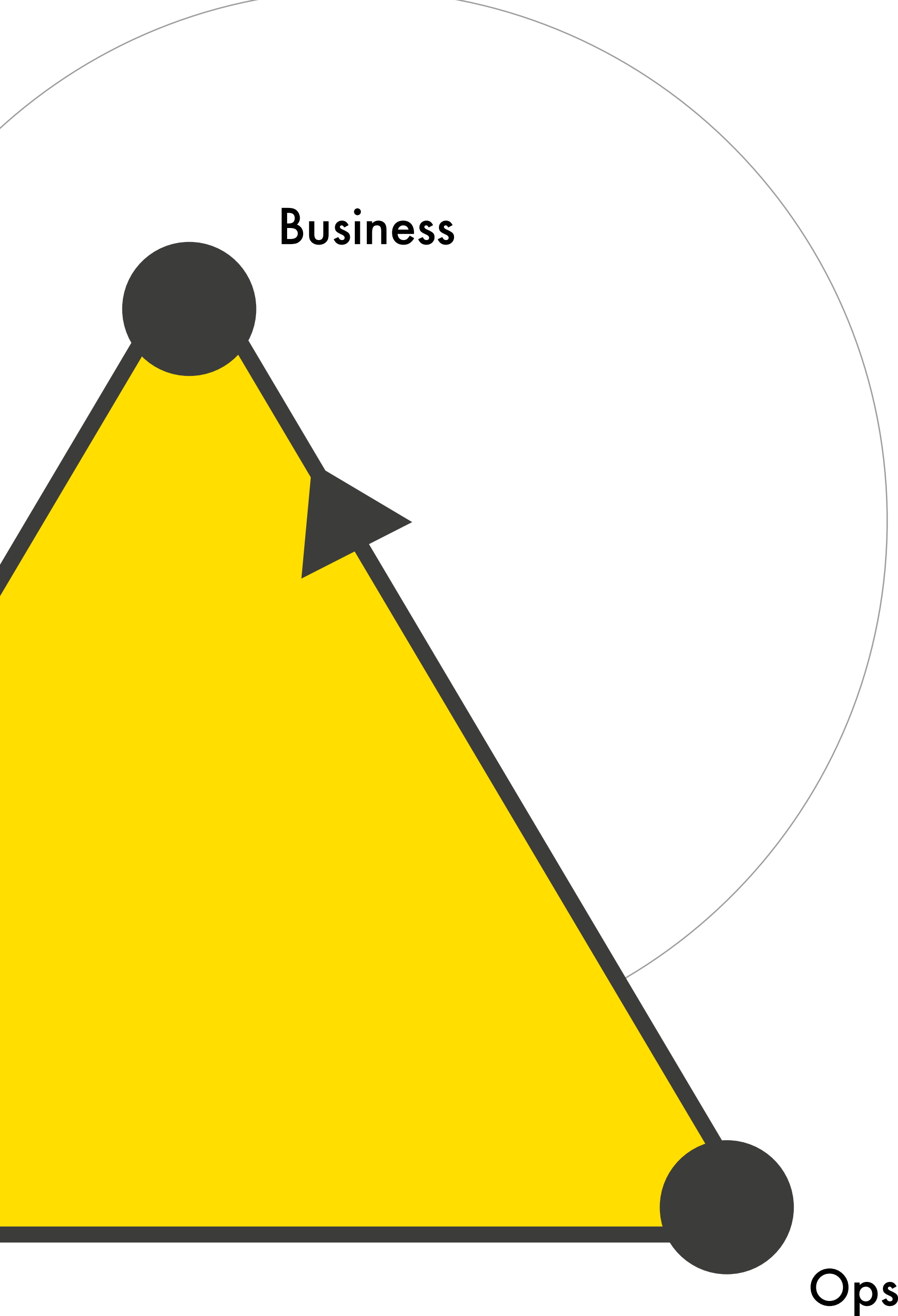
Don't wait for problems,
performance issues or outages.

Go on the front foot and ask
the Business how things are going
and what could be improved.

How's your meal by the way?



CONFIDENCE
& TRUST

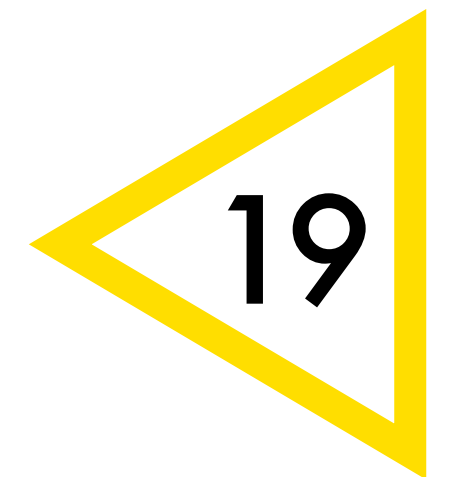


Re-think SLAs and focus on what the Business needs

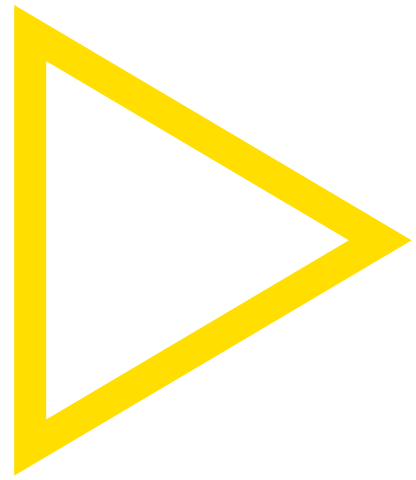
The best SLA is the heart rate of your Business people, not uptime stats or restoration targets.

SLAs are an hallucination of reality – and rarely reflect what the Business actually needs today.

Instead – ask them what makes their blood boil, listen and deliver – the SLA will look after itself.



CONFIDENCE
& TRUST



What next?

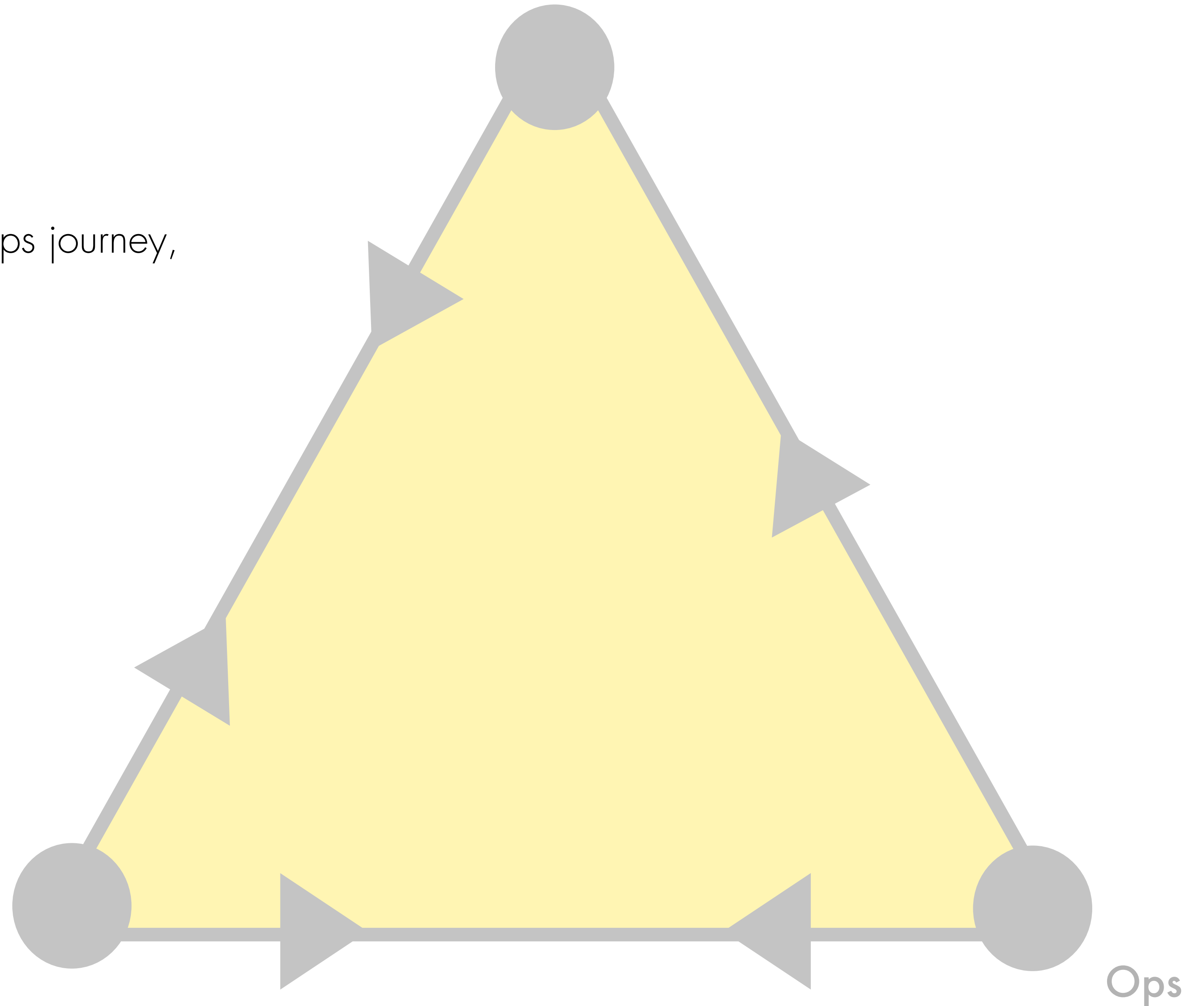
If you'd like help with your DevOps journey,
get in touch.

Visit www.resulting-it.com

START TODAY

Resulting^{it}

Dev



Resulting are independent specialists in
ERP research and consultancy.

Find out more at www.resulting-it.com

Resulting
it

© Resulting Limited 2019

