Making DevOps Doable

How to break five big bottlenecks between you and early DevOps success

pyramid CONSULTING

solutions.pyramidci.com
678-514-3500
info@solutions.pyramidci.com

Contents:

Working software? Not really. 3

Bottleneck #1: **Too big to start with** 6

Bottleneck #2: **Tools over people** 11

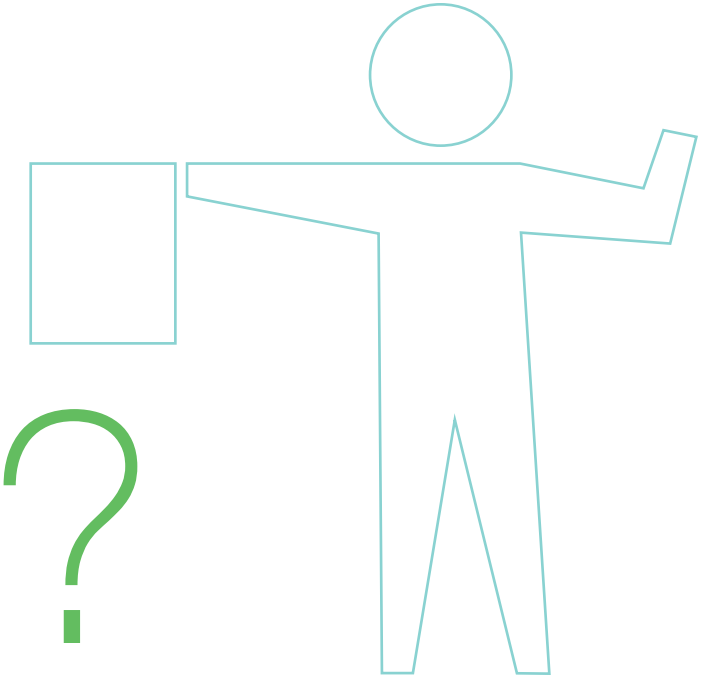
Bottleneck #3: **Ops doesn't want to be there** 16

Bottleneck #4: **The "hardware first" mentality** 19

Bottleneck #5: **Only one map** 22

"When and how", not "if" 24

Working software?



Not really.

Is your new software really “working software” if it runs beautifully on the developer’s machine … survives QA … but then crashes and burns as soon as it’s tossed over the wall to operations?

It's time to cross the last mile in agile development.

According to the Agile Manifesto, "Working software is the primary measure of progress." That's great. But shouldn't "working software" mean more than just tested code? Shouldn't it be software that gets deployed and actually works in the real world?

That's why so many IT organizations are trying to figure out how to cross the last mile in agile development – the gap between development and operations. DevOps really is the final frontier. And there's a lot of confusion around it. To make DevOps doable, you have to overcome five unexpected bottlenecks. That's what this report is all about….

What are high performers achieving through DevOps?

30x more frequent code deployment

50% fewer failures

12x faster recovery from failures

SOURCE: *2013 State of DevOps Report* presented by Puppet Labs and IT Revolution Press

DevOps Bottleneck #1:

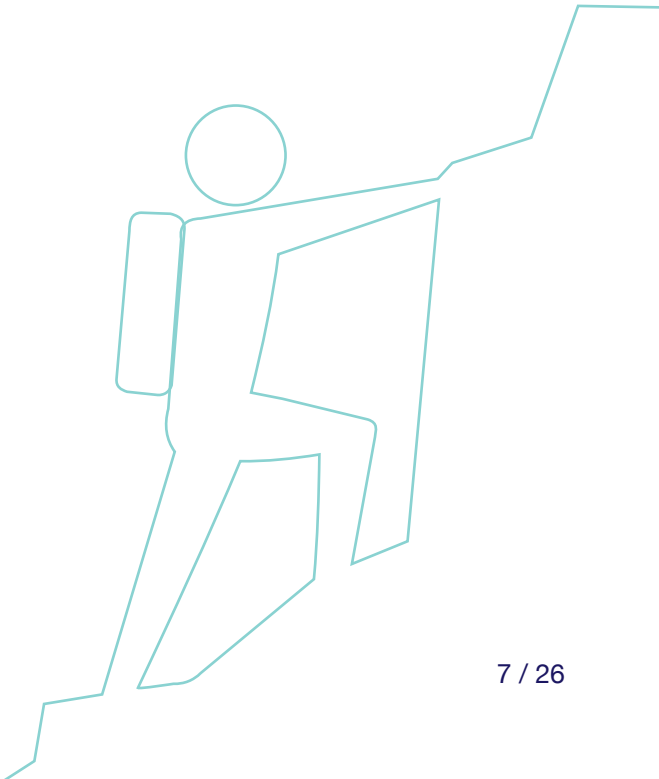
Too big to start with

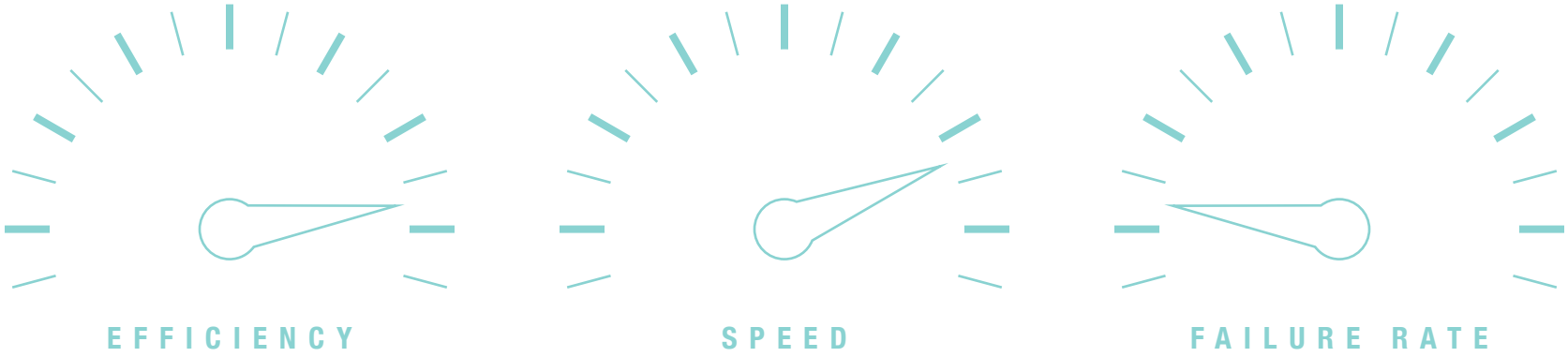
You need to start smaller.

Don’t let DevOps overwhelm your IT organization. DevOps is very doable, but it will require significant organizational and cultural change. **So, don’t try to do everything at once**.

Here’s what we recommend instead:

- **Pick a smaller, shorter initiative as your starting place** – perhaps a new, more innovative project.
- **Of course, it needs to be an agile initiative** – the distance from agile to DevOps is shorter than the leap from waterfall.
- **And think of it as a pilot project** – this lets you get your DevOps feet wet without having to upend your entire IT organization, processes, and work styles.





Smaller also means more specific.

Identify some goals around performance indicators such as efficiency, speed of deployment, and reduction of failures. Make these as clear and specific as possible. (As always, what gets measured gets improved.) Watch your systems logs closely. Identify opportunities for improvement, make the changes, and monitor the results. This gives your team something to aim for and gives you metrics at the end of the pilot to gauge its success.

Get the early win

Start small and closely monitor your goals to:

- **Minimize risks** – by not exposing your whole IT organization to wholesale change
- **Learn by doing** – you'll soon find out what works and what doesn't, so you can begin the process of continuous improvement
- **Demonstrate success** – small early wins give participants in your DevOps initiative a taste of success and shows onlookers how effective DevOps can be
- **Create a model others can copy** – like anything else, DevOps is always easier when you've done it once before or when you can copy someone else's practices

What's a good starting place?

Think about apps for business users. A great example would be the tariff calculators used by call center agents to calculate phone bills.

Or, how about the mortgage calculators used in financial services? These are constantly changing as rates, taxes, and various fees change.

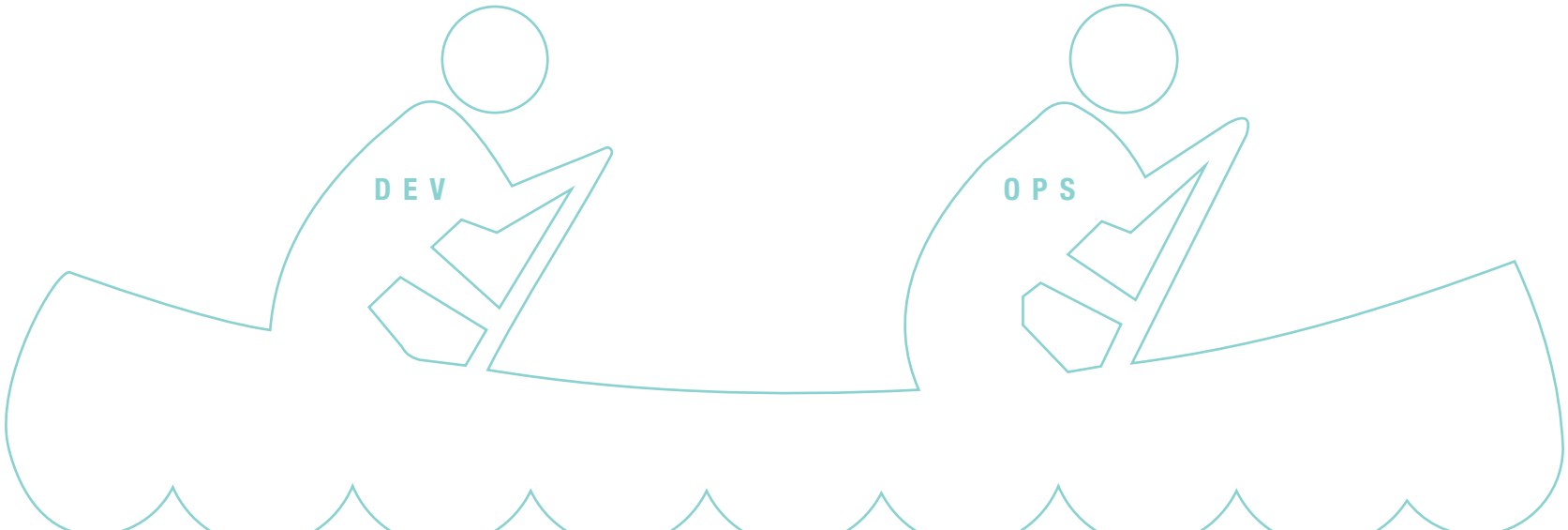
If you build these in an iterative fashion with DevOps, business users will be much more likely to actually use the apps instead of resorting to manual calculators.

DevOps Bottleneck #2:

Tools over people

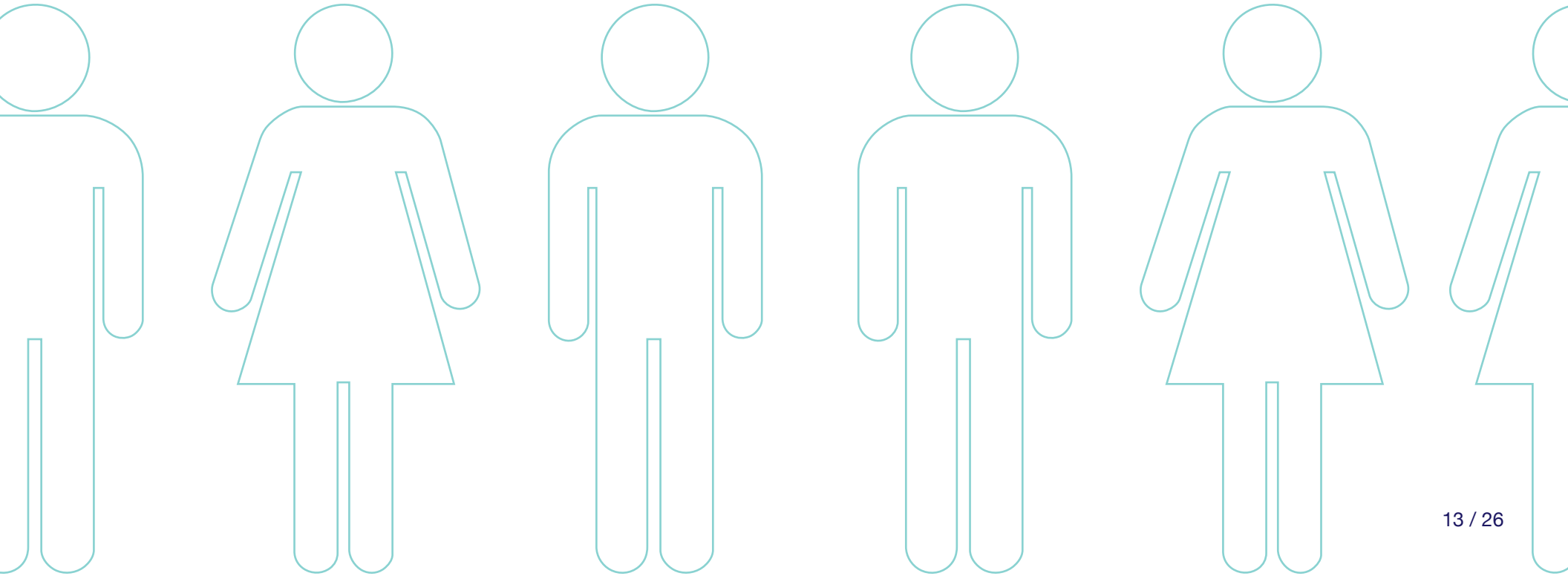
Everyone's obsessing over tools, but it's the people that matter most.

Automation is the key to speed in DevOps, and there are a lot of excellent tools available … from Chef and Puppet to Docker and Dokku. You probably have some of the tools you need already, and there are all kinds of others to consider. But don't be misled – DevOps is not about the tools. It's about people working together. Choosing tools is nothing compared to the cultural and organizational changes needed to get your dev people and ops people on the same page. How do you do that…?

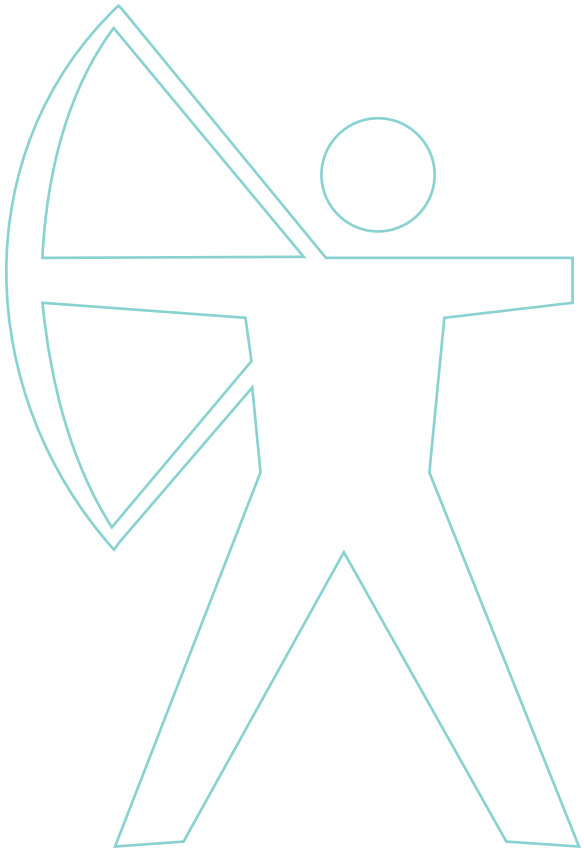


Bring ops into your daily standups.

If your IT organization is truly agile, you've already seen the value of having the product owner in your daily standups. However, product owners don't know all the nuances of operationalizing new software. You need to add somebody from operations to your scrums (or at least, involve them at the end of the sprint and in retrospectives).

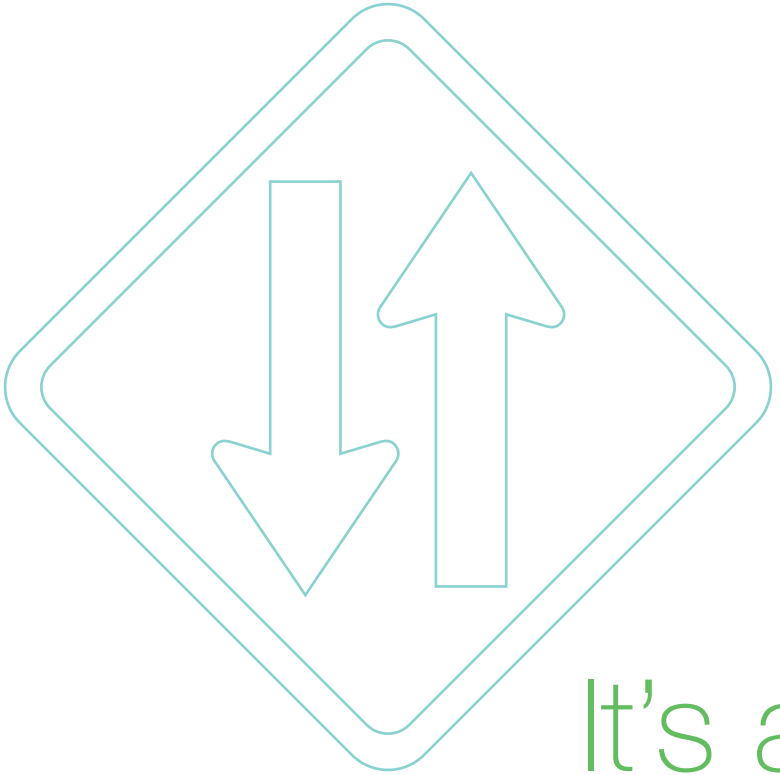


Watch out for turf wars.



What happens when ops wants to have a rep in the scrum? Some people think, “Hey, if the ops people can talk effectively to IT using agile, who needs the business owner to be in the scrums?”

There’s a bit of politics here. Some product owners will say, “I have no ego. Let’s build this together.” But, if you anticipate pushback from a product owner, you can prevent a lot of resistance simply by keeping the owners highly involved in the process and showing them the value of early ops involvement in terms of deployment outcomes (cost savings, time savings, risk management, happier customers).

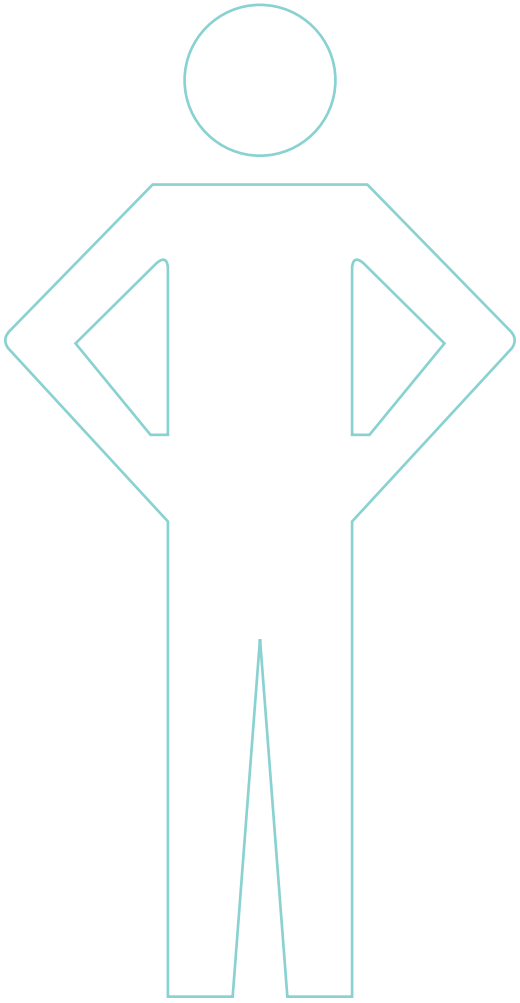


It’s a two-way street.

DevOps goes both ways. You not only want ops people in your scrums, but you also want to start getting your software people into your ops meetings. If the software demands it, you may even want a software engineer as part of your ops organization. The more freely these two groups intermingle, the better your outcomes.

DevOps Bottleneck #3:

Ops doesn't want to be there



Know that ops may be out of their comfort zone.

DevOps is not just a matter of getting the business owner and development team to welcome ops into the SDLC. You may find that traditional ops people aren't all that comfortable being there.

After removing the organizational boundaries between development and operations, you still have to get people out of their comfort zones.

Here are some tips on how to do that….

How to get ops comfortable outside their comfort zone:

Start early	Get operations involved earlier – rather than later – in the process. Not only can you benefit from their input in the planning stages of a new project, but you're letting them get up to speed on the project at the same time as everyone else. You won't be bringing them into meetings later on in the SDLC when they're the only ones in the room who don't already know the project and its history.
Plan joint training and team building	Bring dev and ops people into a joint training session on DevOps. Lay out your goals. Ask questions. Get a discussion going. Include some informal time together by making it a lunch. The idea is to create a team-building experience, so both groups can learn your DevOps objectives while getting to know each other better.
Define the team	Make it clear that ops people are not simply leaving their silos to join the dev team, but that both ops and dev people are leaving their silos to create a new software development and delivery team for this specific project.
Bring in a senior leader	We've seen a company put their most influential VP in as the head of DevOps. It was a smart move. He had the respect of the more change-resistant ops people and was able to get what he asked for.

DevOps Bottleneck #4:

The “hardware first” mentality

Infrastructure can't keep thinking "hardware first".

In a lot of smaller and mid-sized companies, the infrastructure people own DevOps. Typically, these are people who think "hardware first". They're focused on server space, backups, and security. They care deeply about what version of hardware they have and what level of compliance they're achieving. But when it comes to software, they simply expect it to run when they get it.

In a DevOps environment, your infrastructure teams need to make software a part of their thinking and planning. You're not asking them to move up the stack to application development, but you are giving them the opportunity – and responsibility – to make sure the software they get will run on the systems they have.

How do you get infrastructure managers to buy into the change?

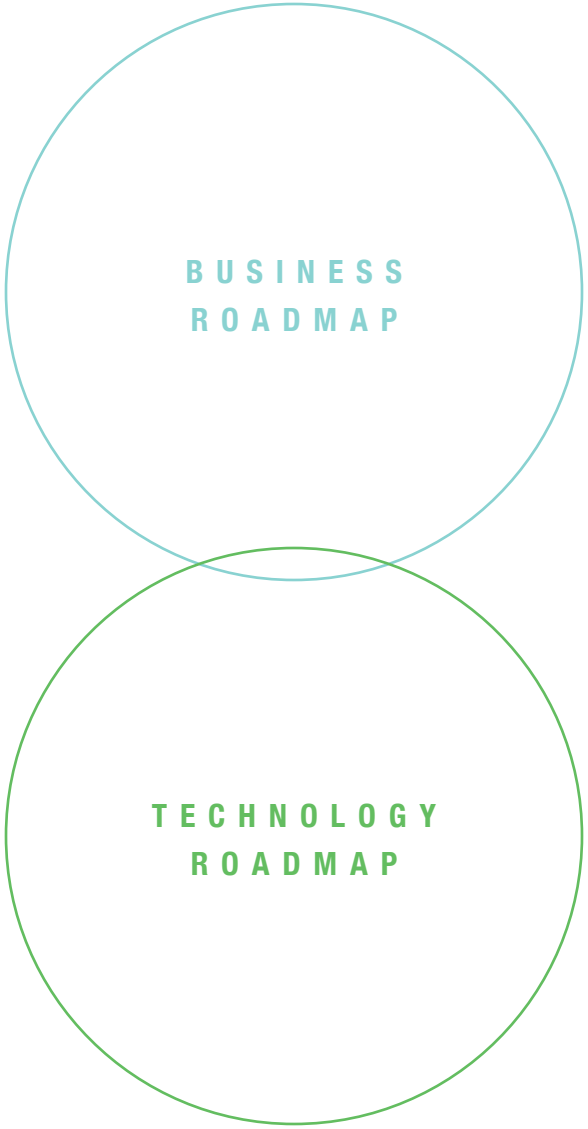
Focus on their pain. They hate getting software tossed over the wall to them that doesn't run right, or even worse, breaks something in another system. A little commitment to DevOps now will make their life much easier later on. As soon as they realize this, these infrastructure managers become your organization's biggest champions of DevOps.

Humility helps.

To make DevOps work, a little humility helps a lot. You want to create an environment where software developers and sysadmins learn from each other and build trust.

DevOps Bottleneck #5:

Only one map

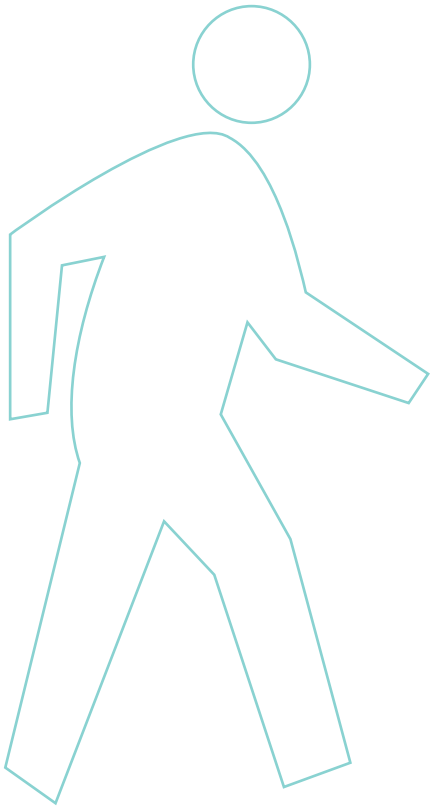


Don’t overlook the power of good documentation – even in an agile environment. It’s especially important now that you’ve got broader DevOps objectives in mind with larger, collaborative teams working on your project together.

Everyone wants a business roadmap, but you won’t hear many people talking about the need for a technology roadmap. That’s a huge oversight. If you only have a business roadmap, you may find that the technology has changed and is difficult to deploy.

Bottom line? You need two roadmaps: a business roadmap and a technology roadmap. The CIO should require both.

“When and how”, not “if”



The shift to DevOps is inevitable. Everyone needs to understand this. Industry watchers say DevOps will have as much impact on IT organizations as Lean has had on manufacturing organizations. We all tend to resist new changes. But, more and more technology leaders are realizing that DevOps is the future of their IT organization. The sooner they adopt it, the greater the rewards.

Do we need a change agent? At what level?

What if we're already far along in the software development process? Is it too late to bring in DevOps? Is there a point of no return?

How closely can – or should – agile and DevOps work together?

Got more questions about making DevOps work? Just let us know.

678-252-1247 or solutions@pyramidci.com

What tools do we need? Which are free? Which aren't? What can we use that we already have?

What about costs? We budget for Dev, QA, BA, and PM – but who pays for the change agent?

We get you past the bottlenecks™

solutions.pyramidci.com

678-514-3500
info@solutions.pyramidci.com

Pyramid Consulting
1100 Atlantis Place
Alpharetta, GA 30022