



# Bottleneck Report on Enterprise Mobile

What's the big bottleneck for enterprise mobile app development this year, and how do you get past it?



<http://solutions.pyramidci.com>

678-514-3500

[info@solutions.pyramidci.com](mailto:info@solutions.pyramidci.com)

# Contents:

- 1 Spotting the bottleneck in enterprise mobile ..... 3
- 2 It's 1995 all over again ..... 8
- 3 Rethinking your mobile CoE ..... 11
- 4 Never build the same component twice ..... 15
- 5 The 10 failure points to watch for ..... 26

Spotting the  
bottleneck  
in enterprise  
mobile

Remember last year's  
bottleneck?

You were saying, "Help!  
I can't get enterprise  
mobile apps out the  
door fast enough."



# Well, this year everyone's saying ... "Help! I need an enterprise strategy."

Here's what happened. Mobile was hot. Your internal resources couldn't keep up. So business stakeholders were getting apps created all over the place. Now, you find that all these independent apps created in separate silos are costing a fortune. With no overall strategy or consistency, they're creating havoc and under-delivering on business expectations. The lack of enterprise strategy is your new bottleneck.

(Oh, and by the way, you still can't get apps out the door fast enough.)



# How do you know you've got an enterprise bottleneck?

- **You've got different developers** in different silos recreating the wheel.
- **There's no strategy**, no consistent brand, no consistent technology or architecture.
- **Too many of your apps are delivering poor ROI** and surprisingly short shelf life.
- **It's all about to get worse**, because you need to get even more apps out the door than you did last year.



# So, how do you get past this year's big bottleneck?

**It's all about creating an enterprise mobile strategy.**

In this Bottleneck Report, we'll look at the three most important things you can do to build that strategy. For each of the three things, we'll walk you through the steps you need to take.

**1** **First, you'll need to build a true Mobile Center of Excellence.** Don't kid yourself. This is NOT the same thing as just having a dedicated mobile team. We'll cover the basics on page 11.

**2** **Next, you'll want to create a reusable component framework.** Why? Because, if you aren't reusing components, your time-to-market will be 70% longer and your costs will be 4x higher.\* We'll show you exactly how to do this in seven steps, starting on page 15.

**3** **Finally, you should check all your individual app initiatives** against the 10 most common causes of enterprise mobile app failure. You'll find these on page 27.

But first, there's one more  
**thing you should know...**

It's 1995 all  
**over again**



# Remember the state of web development back in the mid-1990s?

**It was the Wild West.** There was little process and even less strategy. If you had a friend's sister's neighbor's kid who knew web development, you were in luck. No one thought about enterprise architecture, Centers of Excellence, consistent brand experience, economies of scale ... you just had individual developers doing their own thing. It was exciting. But, it was also incredibly costly and wasteful.

# Mobile is on the same trajectory today, but with one notable exception.

## Enterprise mobile is following the same trajectory the web took.

Every mobile app is treated as a separate, standalone, siloed project with all the accompanying inefficiencies and costs. As a result, organizations are also exposed to all kinds of risks with security, backend integration, compliance, governance, and more.

**However there is one big difference between web dev in the '90s and mobile today:** we have the advantage of hindsight. We've seen how a disruptive technology like the internet matures. So, we can learn from the past and stop repeating the same mistakes.

## We've got a case study called the internet. Let's use it.

What if you could get out of this tactical, reactive approach to mobile app development and use the strategic lessons we learned from the web? Companies that act this year to shift from a siloed approach to enterprise mobile architecture will save money, skip unnecessary growing pains, and leapfrog the competition. Everyone else will get left behind.

So, the first thing you need to do is to build a true mobile Center of Excellence...

Will your organization be ahead of the disruption or running to catch up?

- 55% of internet traffic happens on mobile devices today, and that will only go up.
- People spend 2 hours and 40 minutes a day on mobile devices.
- In 5-10 years, mobile may well be larger than the web itself.



# Rethinking your mobile CoE

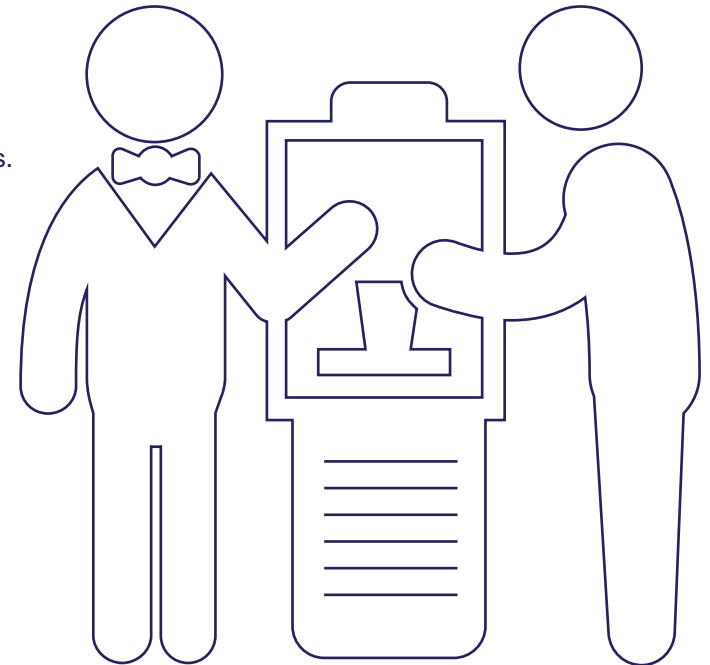
# Your company probably has mobile apps lurking in silos all over the organization.

**The single most important thing you can do about all these issues is to start or strengthen your Mobile Center of Excellence.**

**This is job number one.** It's how you create a robust mobile ecosystem instead of just collecting lots of underperforming independent apps.

## How do you build a true Mobile CoE?

- **First, put the right team in place** – The most important step is to get the right people on the bus. Make sure you include every mobile stakeholder from both the business and technology sides of your organization. Your CoE will fail if technologists are the only members.
- **Then, put the vision and strategy in place** – Once you have the right people in place, you can identify and agree on standard principles, designs, and patterns that will apply across all your enterprise mobile apps. Your CoE will build a strategy roadmap, then ensure that its is socialized and adopted across the organization.



Here's the kicker:

# You don't have a mobile CoE just because you have a dedicated mobile team.

Many organizations think they have a Mobile Center of Excellence, when all they really have is a handful of people who are dedicated to mobile projects. It starts with the right people. But as we've shown, a true CoE must also have a clearly stated vision and strategy for mobile that includes standardized processes, technologies, and frameworks.



## What are the benefits of a Mobile CoE?

You save a fortune on duplicated efforts, ensure a consistent brand and user experience across all your apps, and bring new apps to market much more quickly. With developers spending less time engineering the more difficult aspects of the architecture, they can focus on building the differentiators instead. Plus,...

- **Project approval is streamlined and simplified** – With a healthy Mobile CoE, you have one point of contact and a standard approval process. Making the business case for a new app is also much simpler when you can see how it fits into your company's broader mobile strategy.
- **It's easier to get resources for mobile projects** – If your Mobile CoE is doing its job well, then your new project may cost less than you think. That's because your developers can draw on existing resources, components, and frameworks from the CoE for a faster, easier development cycle.
- **Everyone is finally on the same page** – With marketing, IT and business leaders all on the same page, you no longer run into the competing agendas, overlapping projects, and wasted resources that hinder so many organizations.

Once you have a true Mobile CoE, you can build your path to a reusable mobile framework...



Never build the  
same component  
**twice**

# It's time to stop reinventing the wheel, and start building a library of reusable components.

Let's face it, the closest that most development organizations come to reusable components is just copying code or maybe sending random sections back and forth. There's no organized way to share and reuse the assets they develop within their own mobile projects.

Here's how to do it right...



## Examples of Reusable Components

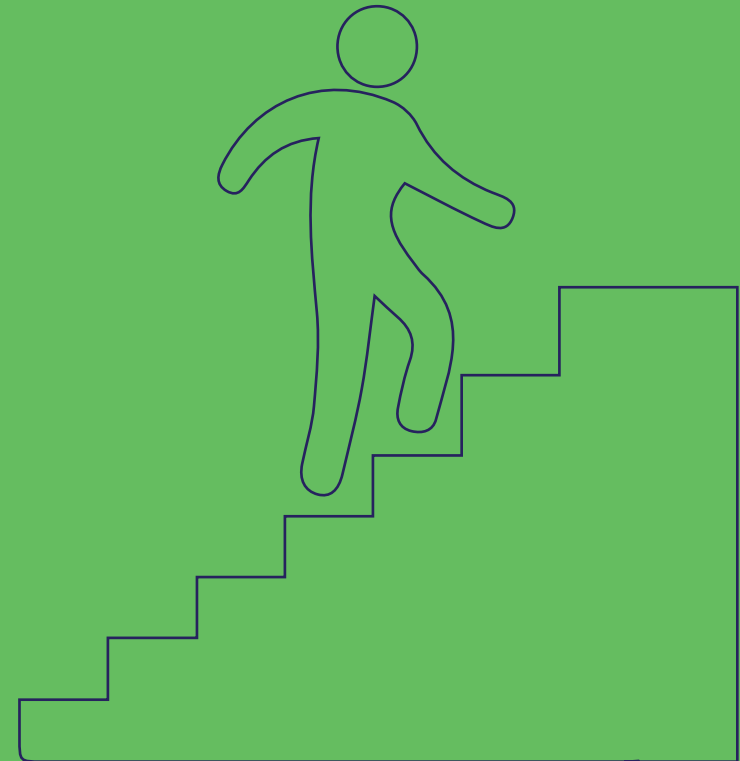
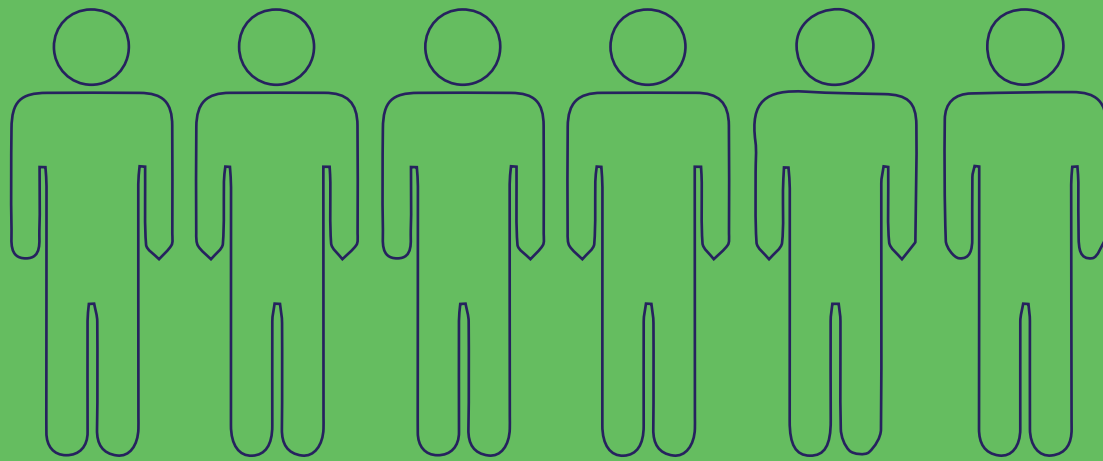
Think about all the apps you need – or will need – that use components like these:

- Camera Capture
- Barcode Scanner
- Signature Capture
- Device Information
- Geolocation
- N
- InApp Browser
- Compass
- Connections
- Contacts





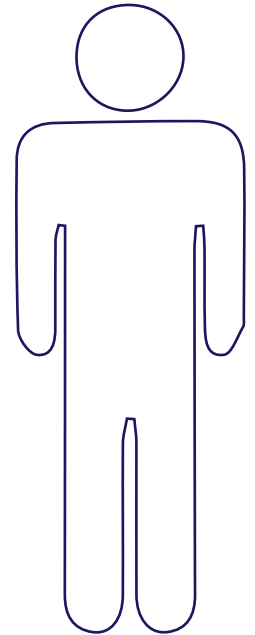
# Seven steps to a reusable component architecture...



# Features

**First, figure out which features you need.**

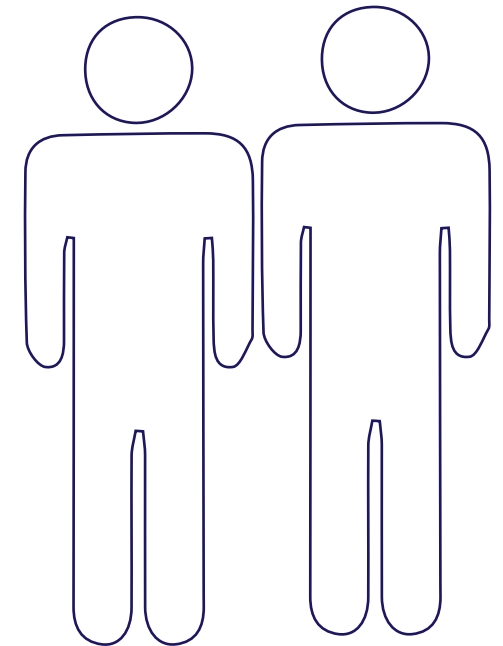
What will they do and why? Identify your objectives. Define the functionality. Prioritize the features.



# MATRIX

**Next, build a matrix for these features.**

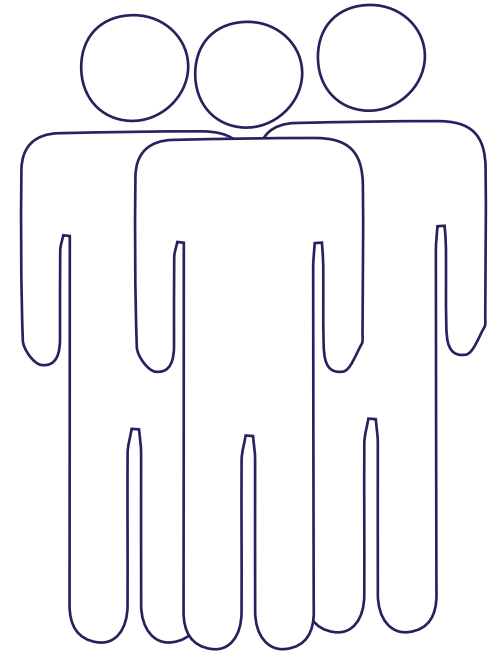
Look at the full portfolio of mobile apps you want to release over the next 12-18 months. Then, break it down into common components that all or many of these different apps will use. This is how you identify your development priorities.



# Description

**Now, create a detailed description of how each component will work.**

Identify how the components will be configured to function differently from one app to another. This is essential for creating reusable components. You can't just copy code and change it to meet different needs. Instead, you want to be incorporating change that is native to the component itself.



# Here's an example...

**Let's say you need an API for signature capture.** Think about how this feature will vary from app to app. What if the color changes from black to blue? Or the pen width gets larger ... or the size of the signature on screen has to shrink? Suppose the image needs to be captured as PNG in one app, but as a JPEG in another? You want to configure all these variations ahead of time, so the component can provide them as different options.

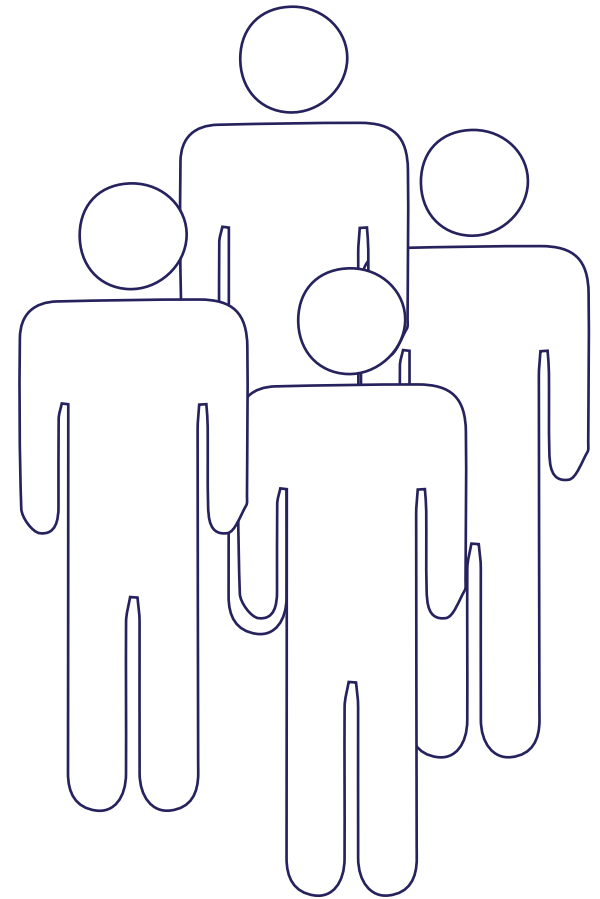
**This means the signature component itself will never change.** In fact, all of your reusable components should never be changed. If you suddenly need signature capture in blue, not black, then your signature capture module should go somewhere else to drive that behavior. You externalize all configuration to a single component that will drive the behavior of all your different reusable components. Developers only change that one configuration component – the other 10, 20, or more reusable components never change.

**This is critical. Individual components never change** – only the configuration changes from one project to another. Why? Because the moment you change one line of code in a component, it's no longer truly reusable.

# Models

**Develop conceptual models to show how your components fit the big picture.**

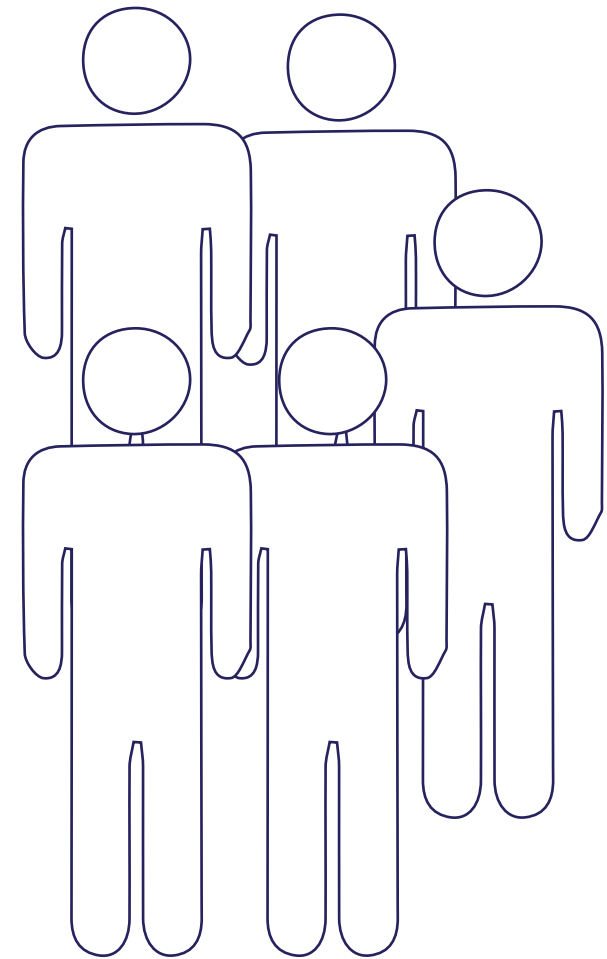
How will you combine components to provide higher level functions? You want to detail your components, their interactions, touchpoints, different layers, and where they fulfill their functions within those layers. This helps development teams see how everything works together.



# Design

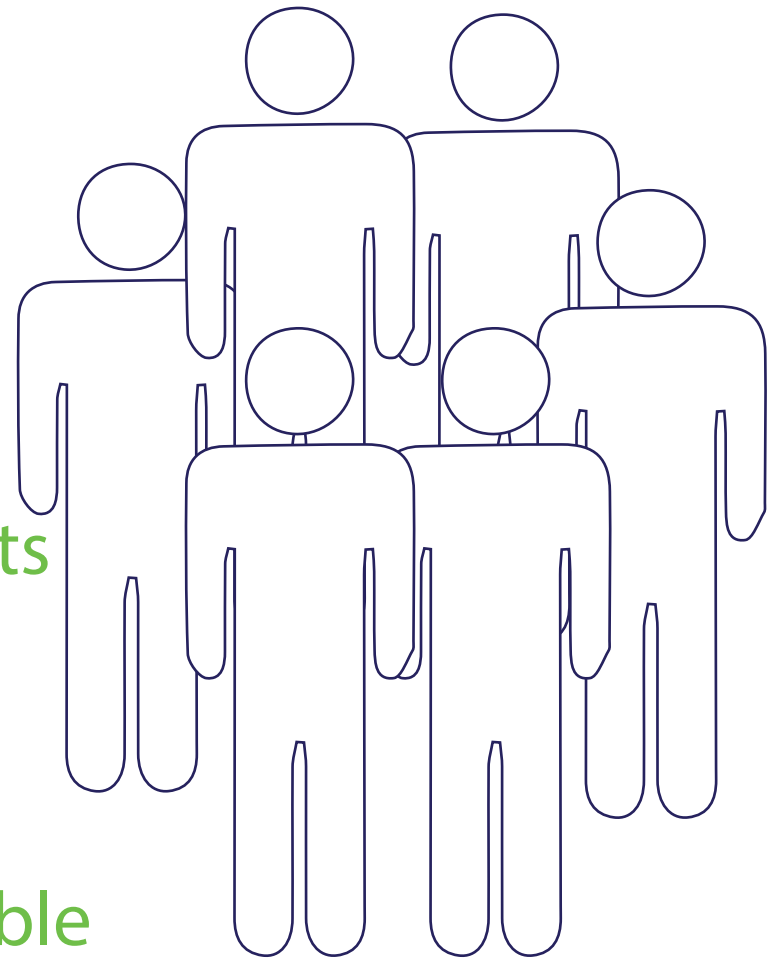
**Now create a physical design from your conceptual models.**

This will drive the actual development. What are the attributes? What about behavior or methods? What functions will be performed? This will drive classes, objects, development. With a logical, documented progression, you can incorporate these components more quickly in other projects.



# Reference

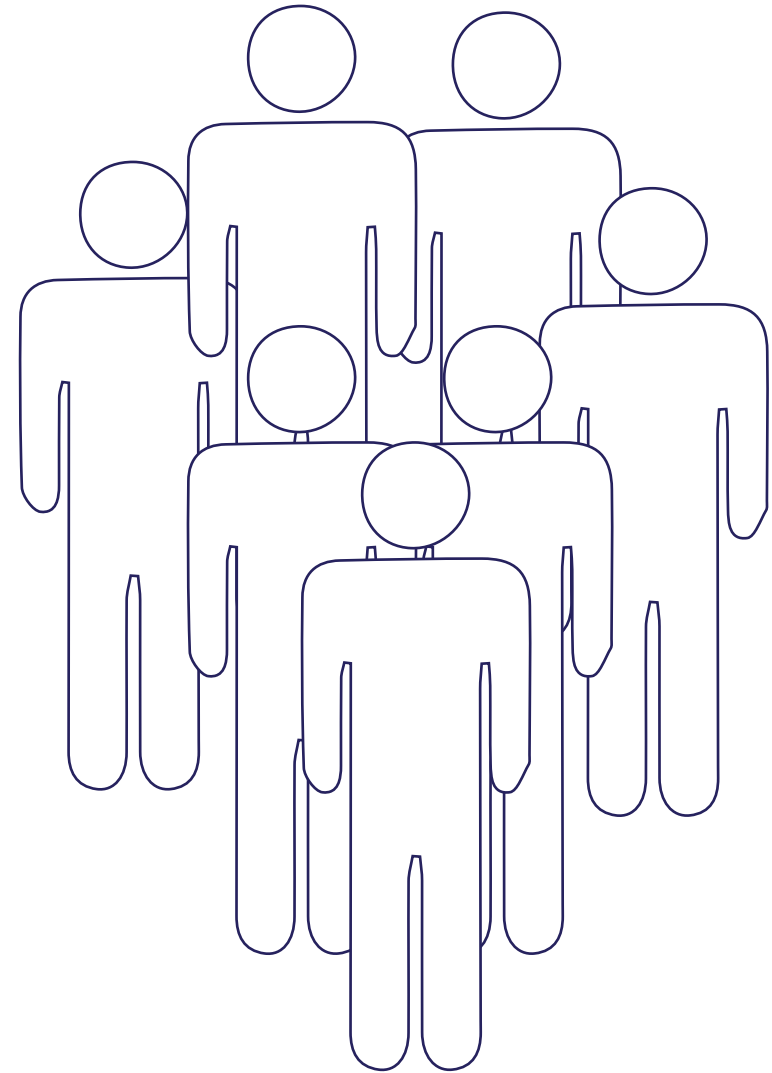
**Build a business reference application.** This app is not specific to any one project. Its purpose is to illustrate exactly how the components will operate. Technologists can see how the app works and the business stakeholders can see the value it brings. Now create a list of configurable components that you can execute on. This makes it concrete.





# Iterate

**Don't stop here.** Iterate and repeat. As your needs continue to grow and change, your mobile CoE will need to work together on refining your architecture and steadily grow your library of reusable components.



# See what's going on here?

**75% of this is about people and process, not technology.** A reusable component framework (1) requires business and technology stakeholders to work together, and (2) requires processes you typically aren't used to following. That's why having a Mobile CoE is so critically important.

Now that you have a framework in place, make sure your development teams check every new project against these 10 failure points...



## **Seven steps to a reusable component architecture**

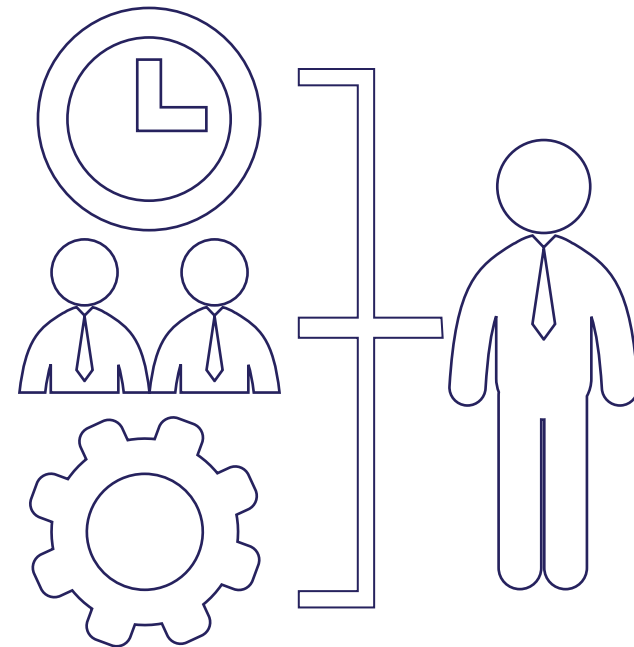
1. First, figure out which features you need.
2. Next, build a matrix for these features.
3. Now, create a detailed description of how each component will work.
4. Develop conceptual models that show how your components fit the bigger picture.
5. Now create a physical design from your conceptual models.
6. Build a business reference application.
7. Don't stop here. Iterate and repeat.

The 10  
failure points  
to watch for

# About 90% of mobile app failures can be attributed to the same 10 reasons - and all of them can be avoided.

**Use these 10 points as a kind of pre-flight checklist** before planning any new mobile initiative, and you will dramatically improve your chances of a successful launch.







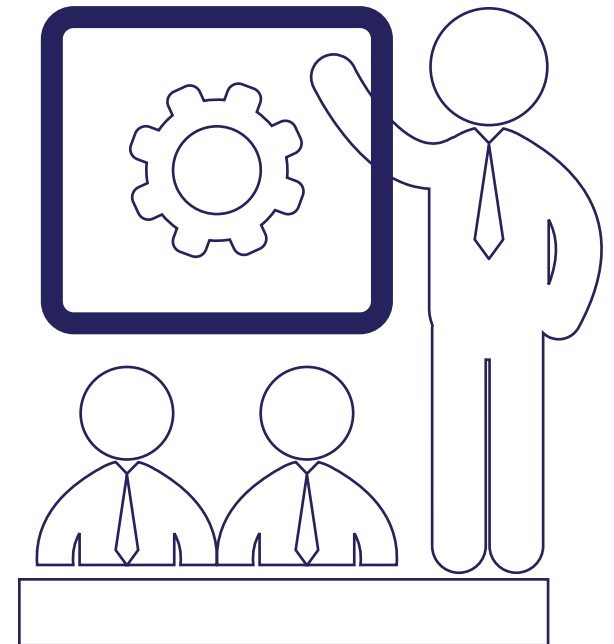
10 reasons. 90% of failures. One checklist. Use these 10 Reasons like a pre-flight checklist. Once you've addressed all 10, your app is ready to fly.

# That's it!

## **Big bottleneck + Enterprise Mobile Strategy = No More Bottleneck.**

Thanks for reading our Bottleneck Report on Enterprise Mobile. Organizations that follow these best practices are seeing much better results from their mobile initiatives. If you have a question about any of these recommendations – or anything mobile – just let us know. We'd love to help.

At Pyramid, we're working everyday to help organizations like yours make the most of enterprise mobile.



We get you past  
the bottlenecks!™



[solutions.pyramidci.com](http://solutions.pyramidci.com)

678-514-3500

[info@solutions.pyramidci.com](mailto:info@solutions.pyramidci.com)

**Pyramid Consulting**

1100 Atlantis Place

Alpharetta, GA 30022