

# Breaking Agile's QA Bottlenecks

Five obstacles in QA testing that every agile team faces. And how to break them.



<http://solutions.pyramidci.com>  
678.514.3500  
[info@solutions.pyramidci.com](mailto:info@solutions.pyramidci.com)



Deciding to deploy agile is easy. **Doing it isn't.** Agile requires continuous testing, but obstacles like bandwidth, money, and deadlines tempt you to stray from best practices – even though doing so will keep the process from working. We'll take you through five QA bottlenecks that every agile team faces. Then, we'll show you how to break them so you can get back on track and create a superior product using a simpler process.



## BOTTLENECK #1

# You don't have time to test.



Bottleneck #1  
You don't have  
time to test.

Bottleneck #2  
You're working towards  
the wrong goal.

Bottleneck #3  
You're using QA  
as a Band-Aid.

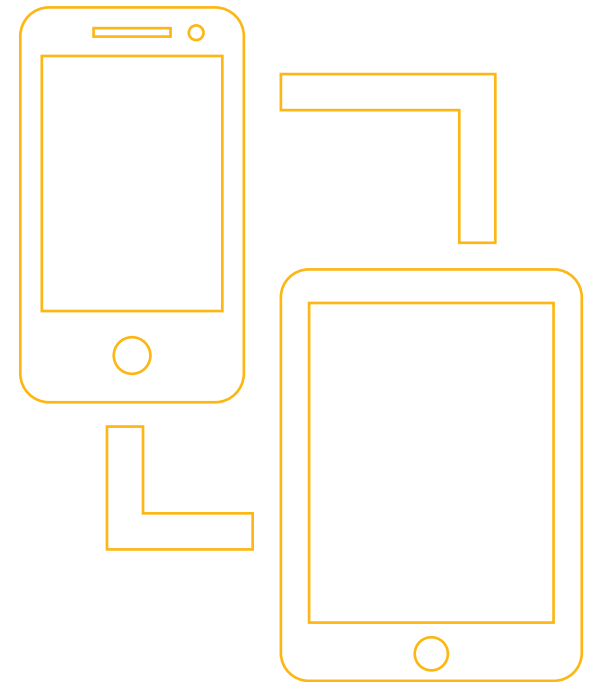
Bottleneck #4  
You're testing  
manually.

Bottleneck #5  
You don't trust  
the process.

# You're rushed at the end of each sprint, so you keep putting off QA until "later."

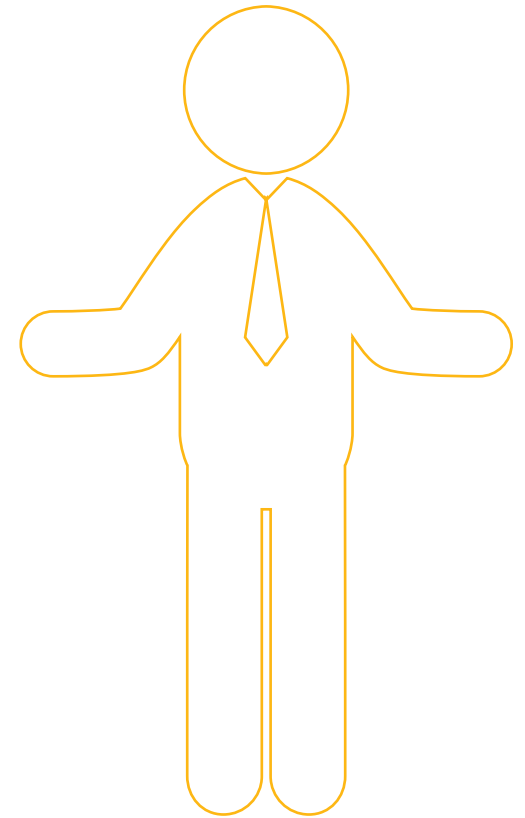
## The problems with "later" are:

- **By the time the bugs are found**, your developers can't remember what they were writing all those weeks ago.
- **Without full documentation** (and often there isn't), you don't have the reference you need.
- **Now QA is backlogged** and will take weeks to finish testing.



## HOW TO BREAK BOTTLENECK # 1 :

Accept the fact  
that you don't  
have time **not**  
to test.



# \$60B

## If you're doing it right, testing won't impede your process – it will simply accelerate it.

Testing during each sprint ensures bugs are caught and taken care of throughout the development cycle. That's partly why you switched to agile in the first place: to avoid the big, painful errors that occur with waterfall implementations. By testing now rather than later, you'll actually save time (and money) by preventing rework down the road.

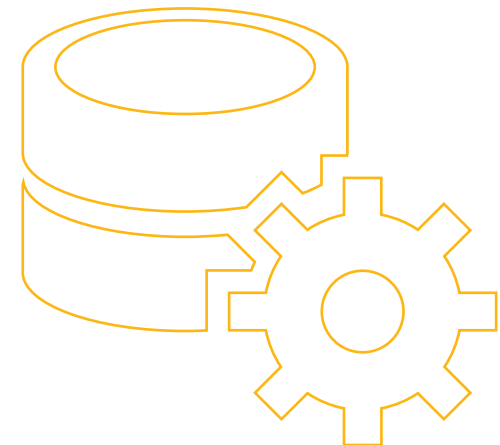
## How integrating QA from the beginning speeds up the development process:

- **Testing can start** as soon as there's something to test, helping you identify and manage bugs early on.
- **Continuous testing keeps** the project's momentum going so you avoid surprises at the end.
- **Quick turnarounds** (due to integrated QA) are the key to powering agile software development.

### The Bottom Line

According to the NIST (National Institute of Standards and Technology), **software defects cost the U.S. economy nearly \$60 billion a year.**

Correcting a bug after release costs more than fixing it during QA, which costs more than catching it during coding.



## BOTTLENECK #2

You're working  
towards the  
**wrong goal.**



Bottleneck #1  
You don't have  
time to test.

Bottleneck #2  
You're working towards  
the wrong goal.

Bottleneck #3  
You're using QA  
as a Band-Aid.

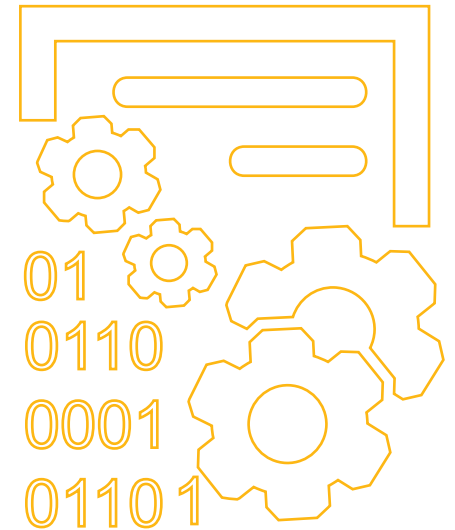
Bottleneck #4  
You're testing  
manually.

Bottleneck #5  
You don't trust  
the process.

# You get points during each sprint for completed code. What could possibly be wrong with that?

Completed code often doesn't do what it's supposed to do. Instead, it creates bugs down the line that you'll have to go back and fix.

In agile, testing isn't a "nice to have" that's done after the goal is met. It's an essential part of meeting the goal.





## HOW TO BREAK BOTTLENECK # 2 :

Realize that  
done means  
more than  
developed.



# Aim for completed functionality, not completed code.


Each sprint should produce software that's deployable, meaning testing has been completed just as it would have been with a waterfall deployment.

In agile, development isn't the finish line. Your sprints should be scheduled to accommodate testing. That way, you keep people from wasting time in the next sprint – touching code that doesn't work.



## Bright Idea

Declare a code freeze two days before the end of the sprint.



# Do's & Don'ts





**Do:**

**Make your new goal tangible** by only rewarding points to tested code that does what the business owner requested (just one reason the business owner should be involved on a daily basis).

**Don't:**

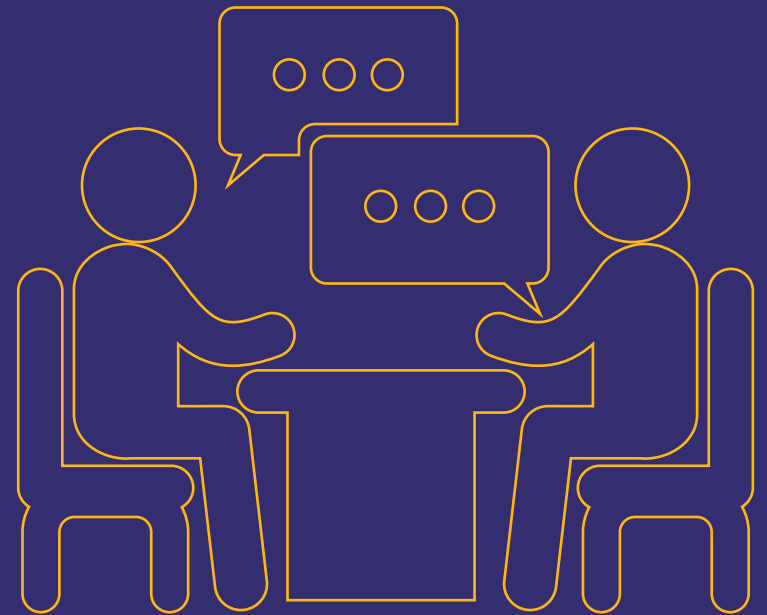
**Find yourself in a situation** where developers are pumping out code without integrating QA every step of the way.

**And  
a tip:**

**If you find it hard to reach your new goal**, determine a level at which agile can be successful and reset expectations. Lengthen sprints to four weeks or reduce your goal from 20 to 10 points per sprint. Just remember to be consistent and only assign points to completed functionality, not code.

## BOTTLENECK #3

You're using  
QA as a  
**Band-Aid.**



Bottleneck #1  
You don't have  
time to test.

Bottleneck #2  
You're working towards  
the wrong goal.

Bottleneck #3  
You're using QA  
as a Band-Aid.

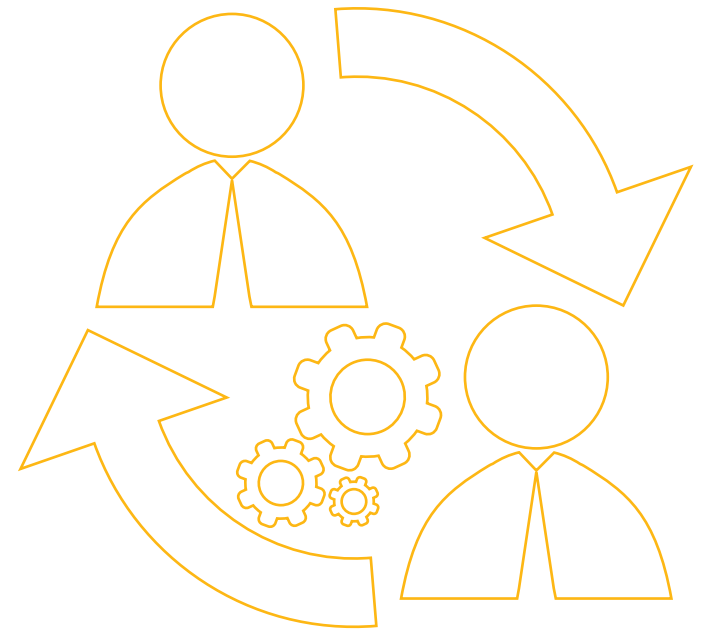
Bottleneck #4  
You're testing  
manually.

Bottleneck #5  
You don't trust  
the process.

# You bring QA in at the end to give software a pass or fail.

## Why that's a problem:

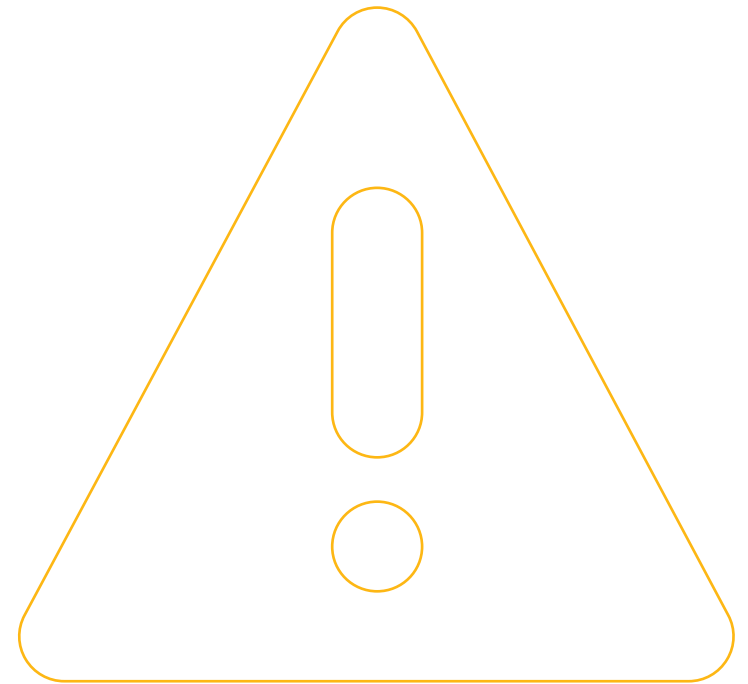
- **It means your testing** is reactive, not active.
- **You're not using QA** to its full potential.
- **QA is no longer measured** by the number of defects found.



## Instead...

## HOW TO BREAK BOTTLENECK # 3 :

Use QA  
to improve  
software, not  
just fix it.



# Don't just ask, “Does it work?” but “How could it work better?”

QA people are capable of more than just bug identification:

- **Testers can help determine** if user stories and acceptance criteria are well defined and if they satisfy customer requirements.
- **They can provide developers** with high-level test cases for user stories before they begin writing code.
- **And they can review unit tests**, even performing exploratory tests on early builds.

Rule of Thumb

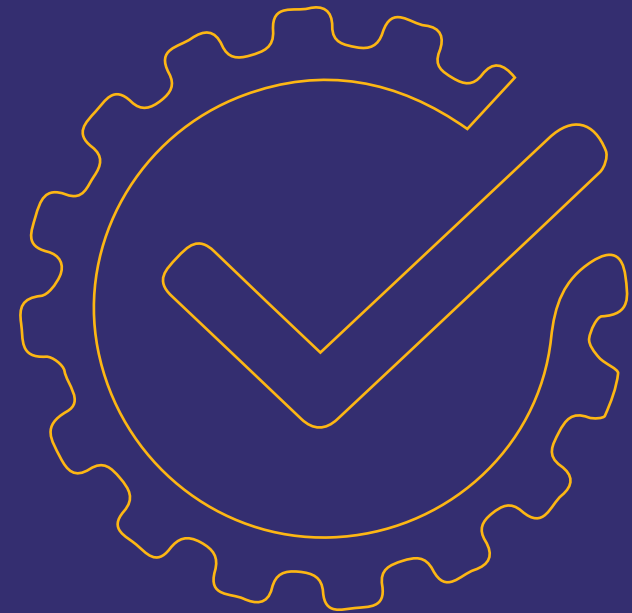
Hire one tester for every two to three developers.





## BOTTLENECK #4

# You're testing manually.



Bottleneck #1  
You don't have  
time to test.

Bottleneck #2  
You're working towards  
the wrong goal.

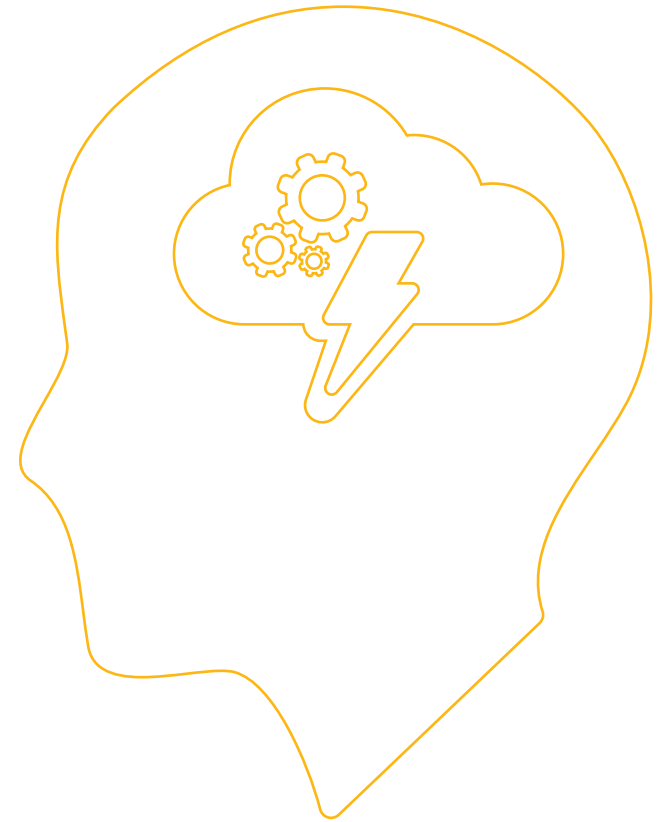
Bottleneck #3  
You're using QA  
as a Band-Aid.

Bottleneck #4  
You're testing  
manually.

Bottleneck #5  
You don't trust  
the process.

# You are using your QA people to test at each stage of development.

That sounds great on the surface, but as long as you're running every test manually, your QA people will never have time to move out of reactive mode or even to fully test the software.



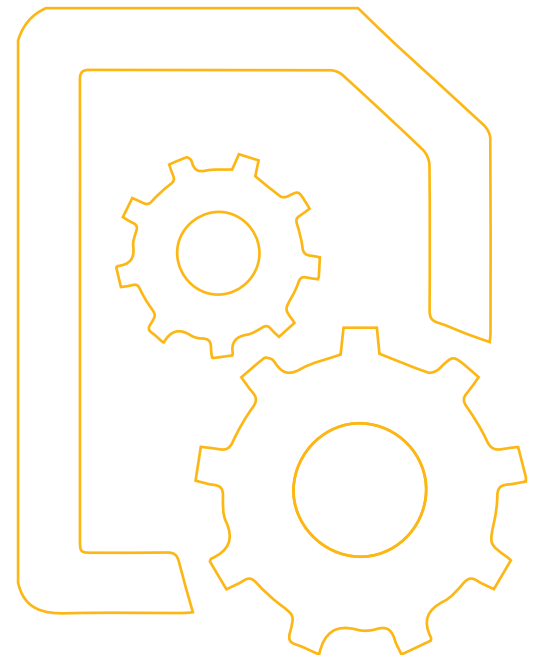
## HOW TO BREAK BOTTLENECK # 4 :

Automate now.  
The easiest bug  
to fix is the one  
you find fastest.

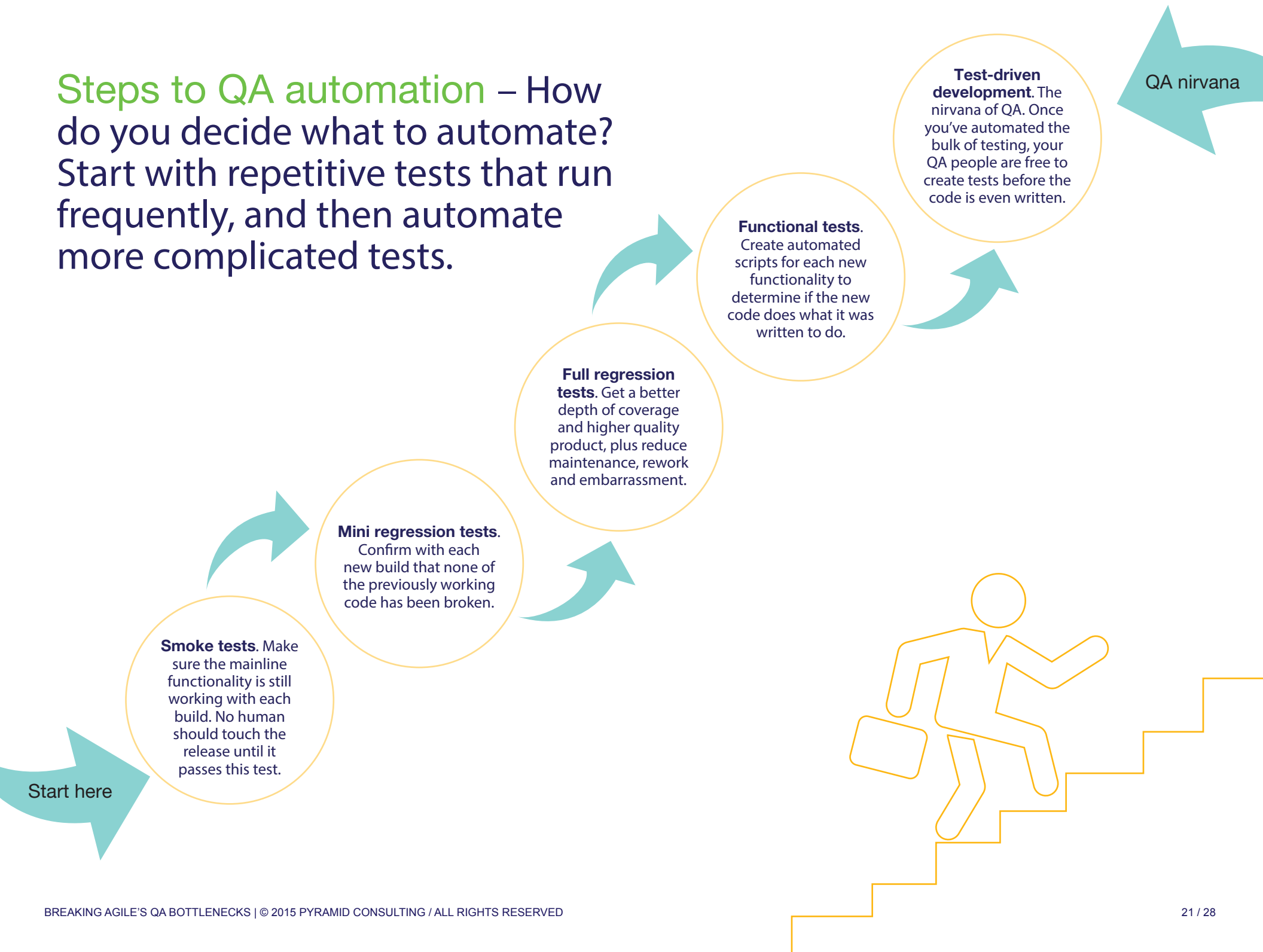


# Automating as many tests as possible enables you to:

- **Find bugs faster and earlier**, correcting mistakes early on and saving valuable time down the road.
- **Free up your human resources** to perform tests that only they can do, ultimately allowing you to test more thoroughly and accurately.
- **Save time, money and manpower** by turning weeks of testing into hours.

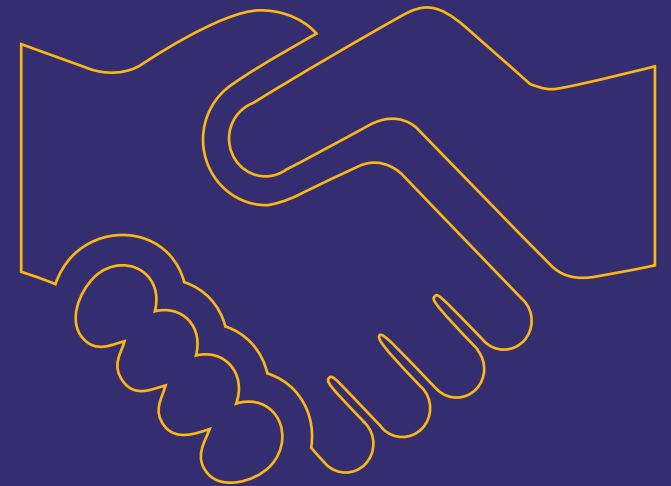


# Steps to QA automation – How do you decide what to automate? Start with repetitive tests that run frequently, and then automate more complicated tests.



## BOTTLENECK #5

You don't trust  
the process.



Bottleneck #1  
You don't have  
time to test.

Bottleneck #2  
You're working towards  
the wrong goal.

Bottleneck #3  
You're using QA  
as a Band-Aid.

Bottleneck #4  
You're testing  
manually.

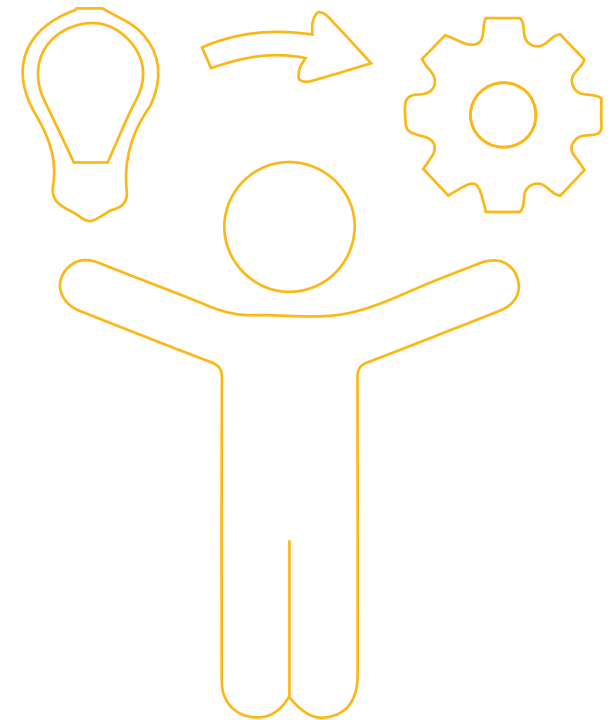
Bottleneck #5  
You don't trust  
the process.

# Is your agile implementation just waterfall in disguise?

Many teams are treading the no-man's-land between agile and waterfall without gaining the full benefits of either.

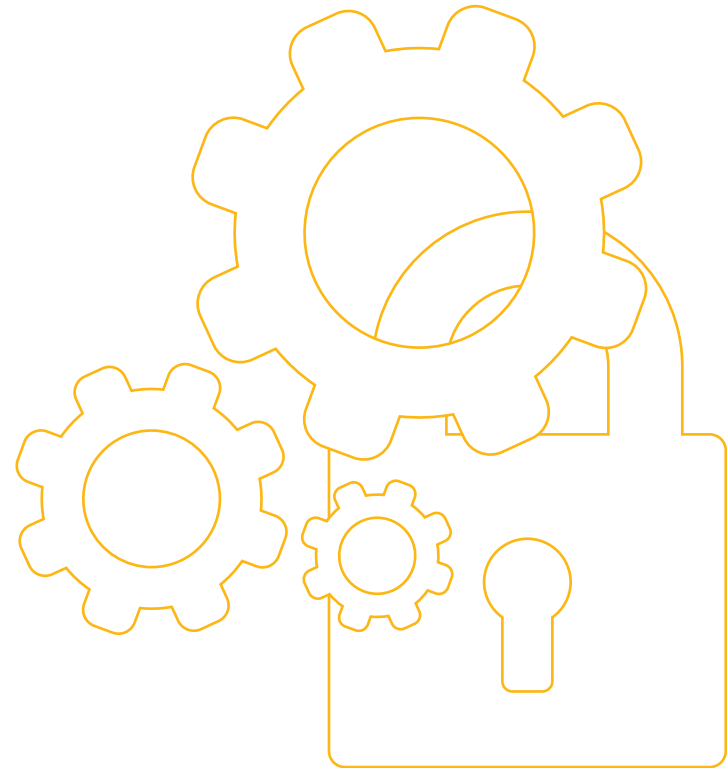
## See if you recognize these red flags:

- Changes are difficult to implement.
- Days go by without you seeing the business owner.
- Your QA people look like they've been hit by a truck.
- You find bugs in software that's already gone to market.



## HOW TO BREAK BOTTLENECK # 5 :

Agree that  
if it's not  
tested, it's  
not agile.





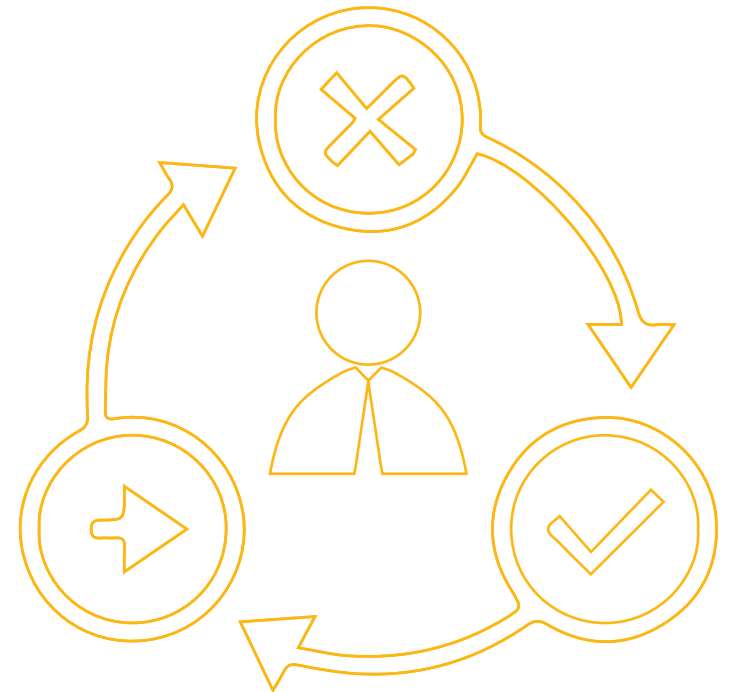
# Code without QA may look like agile, but it won't succeed like it.

The agile process is about fast feedback, continuous changes and group ownership. If QA isn't intimately involved in the development, these aren't happening. To review:

- **Involve QA from the beginning** and integrate it into every team.
- **Automate as many tests as possible** and use your human resources to improve functionality, not just find mistakes.
- **Count code as complete only when** it is fully functional and does exactly what the business owner requested.

agile + QA = :) )

If you can break past  
these bottlenecks,  
you're on your  
way to smoother  
development.  
Now that's agile.



# Got more questions about QA testing? Just let us know.

Our QA experts will be happy to advise you on any issues related to testing in an agile environment – based on our years of experience working with organizations like yours around the globe.

We can also offer you the resources of our Pyramid QA Center of Excellence. It provides state-of-the-art testing products, tools, and custom frameworks as well as QA teams in the U.S. and offshore.



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