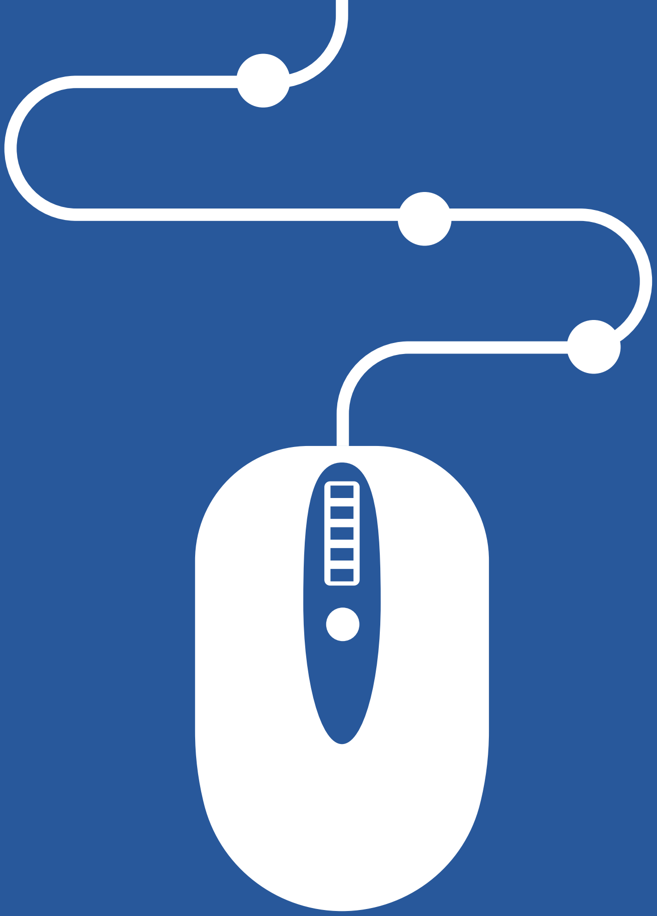

Go To

PRODUCTION 4 WEEKS EARLY

Why Get Your New Production System Up and Running 1 Month Before Going Live?

windward studios

Go To Production 4 Weeks Early

Why Get Your New Production System Up and Running 1 Month Before Going Live?

When you have a new production system, you want it to be perfect before you start promoting it in your organization. Indeed, you don't want to be bombarded with complaints about bugs and issues. As we all know bugs that reach the end-users are always expensive and above all embarrassing to fix!

This is why testing in production or more commonly known as TiP (testing-after-deployment) is crucial if you want your system fully-functional and free of any issues.

In fact, it is just the TiP of the iceberg (no pun intended :)

Testing in production can boost confidence that your new system can handle everything that a live environment may bring. When you test during production, you can see how it would perform if new codes are introduced in the real environment among other things.

In fact, getting your new production system up and running a month prior to it going live is a smart idea.

Read on to Know Why →

```
require File.expand_path("../../config/environment
# Prevent database trunca
abort("The Rails environment
require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver =
Category.delete_all; Category
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires below this line.

# Requires supporting ruby
# spec/support/ and its
# run as spec files by default.
# in _spec.rb will both be
# run twice. It is recommended
# end with _spec.rb.
```

Potential Risks

While testing-after-deployment (or TiP) brings forth a myriad of benefits, there are potential risks to it. Depending on the type of application you have, you could run the risk of experiencing bugs when you are doing testing at this stage.

Here are some of the common threats and setbacks that you might encounter:

There is a possibility that you could expose undiscovered vulnerabilities to the public, thus making it easier for potential hackers to compromise your whole system.

There is also a high chance of losing data especially when you don’t have a recovery or backup plan.

While testing is ongoing (continuous), end-users could have a bad experience. This will most likely lead them to leave your services for good.

You can find some conflict, incompatibility or other issue when you have everything installed. Sometimes this occurs on only one of the servers which generally is very difficult to resolve.

Doing It Right

Even with these major threats, you still think that testing is more beneficial than it is risky. If you are one of those hard-headed developers who advocate that testing is a must, then you better make it right.

Take a look at these tips, so you don't mess it up.

1. Test in Layers: The production environment is divided into phases. Sometimes, you will have a test system running side by side with your production platform. Other times, you want to run your tests on the live system itself. It is important that you layer your tests so you can actually analyze the behavior of your system in different situations.

2. Test When Traffic is Light: We already mentioned that when you test in production, you are accessing the system simultaneously with your customers. To avoid messing things up for the majority of your customers, try to find a time when most of your customers are not using the system. Or, you can do your customers a favor by giving them a fair reminder — probably days before your planned testing.

3. Focus on Metrics: The primary goal of testing is to improve the user experience. While your tests are going on, take a look at your metrics. This allows you to determine if the tests actually affect the current user experience of your customers. If it actually does, do not hesitate to shut down the test.

The Big Question — Is It Worth It?

Sure, TiP should become apart of your QA. After all, this is where you actually test your system, weigh its performance when it is already handling high traffic. But, the risks you expose your system to actually outweighs the benefits if you make post-deployment testing your only testing method.

For instance,you would obviously be accessing your application simultaneously with your users. This means that your users might find bugs and glitches first before you do.

Oops...there goes your good customer experience down the drain!

OR

Get Your Hands Dirty in the Laboratory —Prepare for Pre-release Testing

4. Put Your Best Foot Forward — Allow Opt-in Experience The best thing that you can do to minimize the impact of testing to your customers is to offer them the chance to become test subjects for a beta-testing of new and improved features you plan for the system. Consider a pre-release testing. This way, you can lay out everything to the users who chose to opt-in for the experience and tell them that inconsistencies may be present as it is a testing stage. You can do this at least a month before it actually goes live.

Here's How to Ensure Superior User-experience:

Here's what we suggest — **remember point #4 from above?**

You can play it safe and do all the testing and running before it goes live.

Get it up and running and throw in as many real-world-scenarios as you and your team can think of (create the best possible simulation you can to test your new production system to eliminate any future failures).

Have a few end-users test it. See how the application works and ensure you have a disaster recovery plan that you can implement if something goes awfully wrong.

This way, you can iron out any wrinkles and your end users will not experience any problems the moment it goes live and they start using your application. Give yourself a deadline. Aim for a completion of the testing process at least a month before you plan to go live.

In conclusion, while it is important to make continuous testing or TiP a part of your QA process,you should not make it the only testing method you have. It can be harnessed as a way to spot and get rid of bugs. This is what the 'Help' and 'Ask' pages are for. If you are receiving feedback from your users about certain issues, then it is time you do TiP.

In short, get your production system up and running at least 3-4 weeks before it goes live and test it thoroughly before the final roll out.

About Windward Studios

Windward Studios makes document generation software that allows companies to c**onnect document templates** to **multiple data sources for high-volume output** in a variety of formats. Our design tools offer unmatched flexibility, ease of use and control, and our output engines can be embedded in .**NET**, **Java** or **RESTful** applications.

Visit **www.windwardstudios.com** for more information or **START A FREE TRIAL →**