

Continuous Automation, Continuous Testing

**A Powerful Approach
for Scaling Automation**



Are you running into roadblocks in scaling automation?

This paper will tell you why it's happening. And what you can do about it.

Is your automation program hitting issues like these?

Maintenance

Your team is spending too much time maintaining existing robots and not enough time building new ones.

Friction

You can't deploy as many automations as you want to, as fast as you need to.

Scope creep

Constant changes in project scope and capabilities requirements push your deadlines out, again and again.

Quality

Manual testing your workflows is a big headache and drags down productivity.

If these are the type of problems that you're dealing with, don't despair. Many automation programs face these issues as they attempt to scale up.

This paper will give you a path to scaling success.

Let's get started.



01.

How continuous automation development and **continuous testing** can help you scale your automation programs

The pitfalls of waterfalls

If you're having problems scaling RPA, it just might be the process you're using

Are you using a waterfall process like the one to the right to develop and launch automations?

If so, you're not alone. Many automation programs start out with this step-by-step approach. And that's fine if you're doing a small number of relatively simple automation tasks.

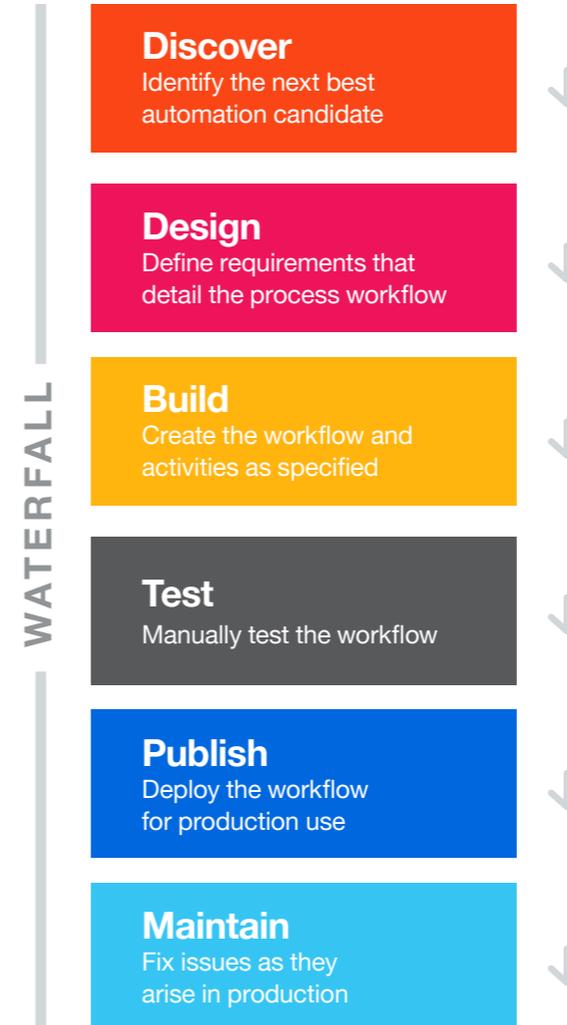
Unfortunately, as you take on a greater number of more complex RPA projects, problems start to surface with a waterfall approach.

- Lack of a consistent process creates bottlenecks and staff underutilization.
- If requirements change or issues are found after you've started building, your deadlines may suffer.
- Insufficient testing leads to failures and higher maintenance rates

That's why more successful, scaled automation programs use a different approach they've adopted from modern software development.

It's called "continuous automation development."

TYPICAL STEP-BY-STEP WATERFALL DEVELOPMENT APPROACH



Did you know?

Waterfall will work for those exploring RPA, but **adopting an agile, continuous approach will accelerate your outcomes** while boosting robot quality.



Move to continuous automation development

Take your lead from modern software development

Twenty years ago, software developers faced issues similar to those confronting RPA developers today. They realized they were bumping up against the limits of the waterfall approach. **So the software community adopted a new process: continuous software development.** You may know it as Agile (including specific approaches like Scrum or Kanban) or by its newer cousin, DevOps.

But whatever you call it, it's based on the same concepts. Continuous software development relies on an iterative, incremental approach that prizes collaboration, fast releases of working software, responsiveness, high adaptability, and ongoing improvement. **Here's how it works.**

Continuous Software Development: a modern approach to software development

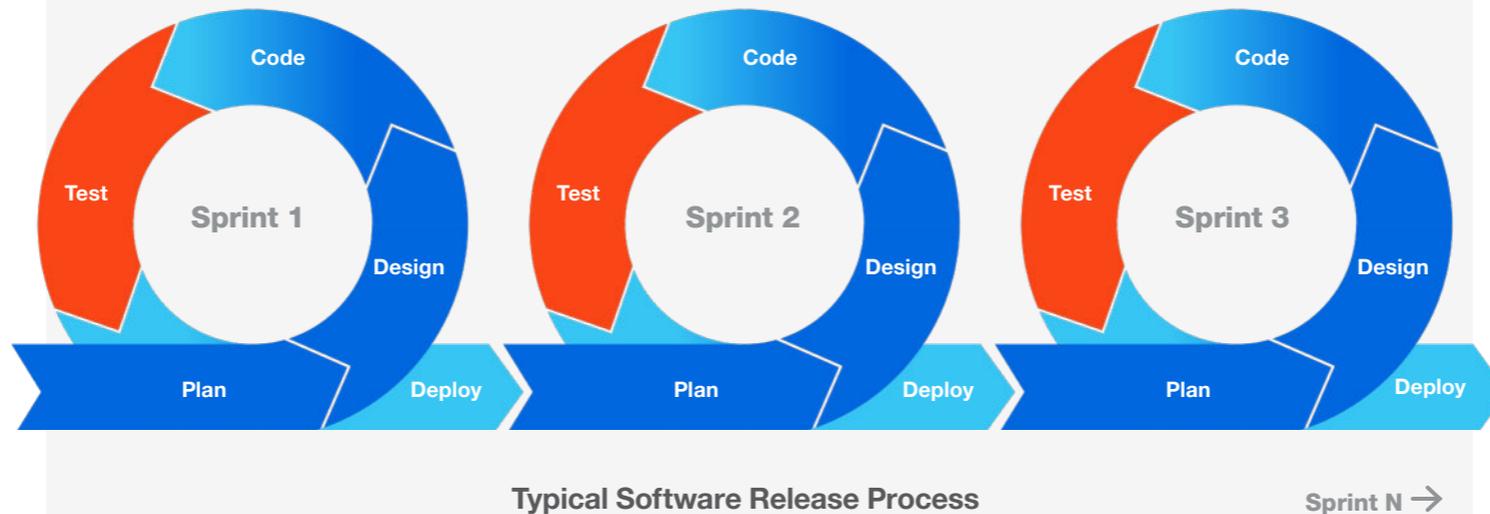
Working closely with end users and stakeholders in a collaborative group, developers gather initial requirements.

Moving in short time-boxed iterations, they plan, design,

build, and test an initial version. Iterations continue until a version is deemed production-ready and is deployed in a release

Once in production, software is continuously monitored and

tested for any regressions. Any issues found are proactively addressed before they impact production and user feedback is incorporated into subsequent releases.



This approach supports software development at high scale. It radically speeds up release cycles while improving design flexibility, team productivity, and overall quality.

These are key reasons why today most software development teams follow a continuous, rather than a waterfall approach. RPA teams can too. (After all, isn't RPA development a form of software development?)

But you need some critical capabilities to be able to do so. Chief among these is **continuous automated testing.**

Did you know?

Software teams that adopt a continuous approach, such as DevOps, deploy 200X more frequently while reducing failure rates by 3X and recover from failures 24X faster.

(Source: 2016 State of DevOps Report)



Continuous development relies on continuous testing

An automated testing capability is critical

Continuous regression testing will tell you if any new features or break fixes will impact software that is currently working in production.

It's safe to say that, without rapid, reliable, continuous testing, continuous development would grind to a halt.

But think of the challenge you'd face if you had to do all this testing manually. You'd need massive testing and QA teams to handle this never-ending, constantly growing burden.

Software teams that adopted a continuous approach came up against this testing bottleneck early on. So they sought out a new solution to address it: **automated testing tools.**

With automated testing tools, test developers write scripts once, and re-execute this test script over and over. These tools map tests to requirements, manage test cases, organize scripts, execute tests, capture and report results, and track defects.

Automated testing tools have been around for almost 20 years, and they're used by virtually every software development team that follows a continuous development approach.

To adopt continuous *automation* development, RPA teams will also need to have some form of automated test capability. **And in fact, RPA developers need automated testing even more than software developers do.**

Here's why.

Testing occurs continuously throughout the software development process



Continuous development generates a constant stream of new iterations that must be thoroughly tested before release to ensure they will all work together.

And that's just the start of all the testing to be done.

That's because a critical aspect of continuous development is its feedback loop—the connection between development, QA teams, and operations that's at the heart of a DevOps approach. And continuous testing of the software that runs your business is a critical point of feedback.

Continuous testing is particularly critical for a large-scale automation program



Constantly changing applications and environments demand ongoing monitoring

Robots often work within applications—some third-party, like Salesforce and SAP, and some home-grown.

So when you create a robot, you build it to interact with an application’s particular UI, its specific run-time environment, and the like.

But those things change all the time, often without notice. Most business have hundreds of applications that may be updated weekly, daily, or even hourly.

We’re talking a constant slew of changes, any one of which can disrupt a robot’s flow. It’s far better to find and fix potential

disruptions before they happen, rather than to have to react with “all hands on deck” under crisis mode.

That’s why you have to continually test automated business processes proactively to identify issues before they impact production.

Did you know?

“In today’s software-driven climate, the best tech companies—Facebook, Amazon, Netflix, Google—are releasing software updates thousands of times a day.”

– Adil Aijaz
DevOpsZone, 2019



02.

How to implement continuous testing to support continuous development

Make continuous testing an integral part of building robots

To implement continuous testing as part of a continuous automation development approach, you'll follow a four-step process. The most important thing to understand about this process is this: **testing capabilities are part and parcel of the robot development**

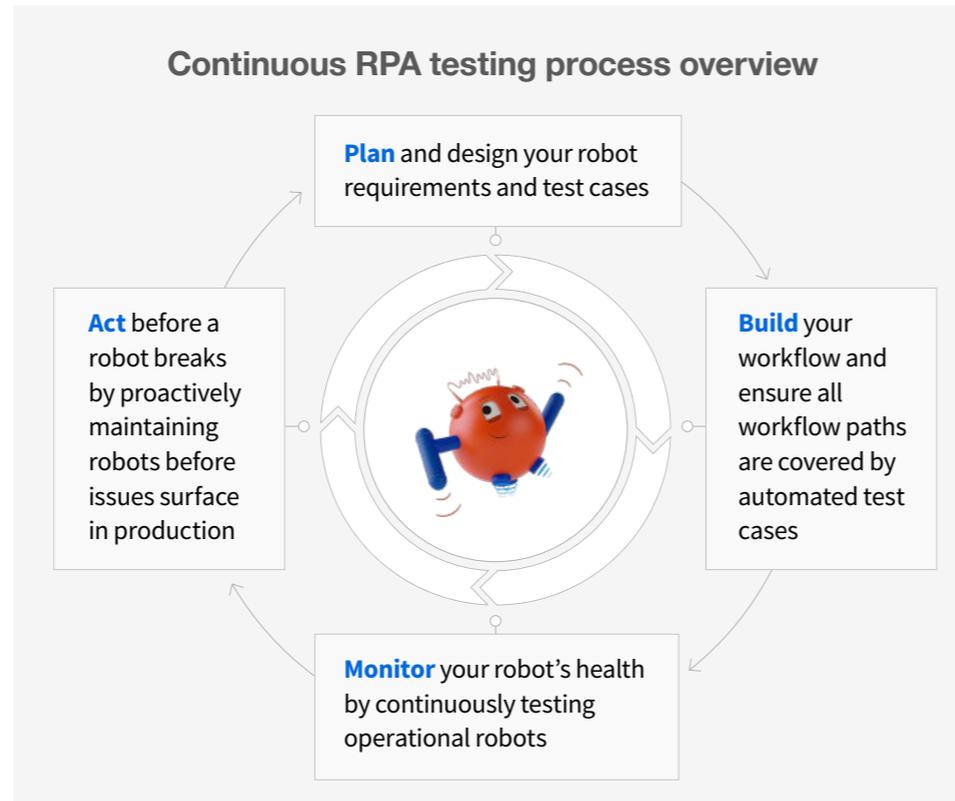
process. Unlike a waterfall process, testing isn't a separate, discrete step; rather, it's a necessary part of a complete robot package. No robot should be published unless it also has an accompanying set of automated test cases for its workflows.

You'll start the process by **planning and designing the robot and the test cases at the same time.** Next, you'll **build the test cases simultaneously with the workflows themselves.**

These test cases will be stored within the workflow itself. They'll first be used to certify robots before initial deployment. Subsequently, they can be dispatched to continually **monitor robots' health** via a set schedule, on demand, or both.

If this monitoring identifies workflow problems, you can **act proactively to fix issues** before real-life production problems arise.

On the next several pages, we delve into this process in a bit more detail.



Did you know?
Best practice is to add tests as you build your robot's workflow.



Create automated tests while you build

Use these capabilities to test your robot's workflow prior to publishing

Under a continuous automation development approach, testing isn't a single, discrete step to be completed before you release something. It's an ongoing activity that's part of the process of building robots.

Plan and build workflows and workflow tests simultaneously. It's a package deal: every automated process should include its own group of automated test cases that support its continuous monitoring and maintenance. Part of planning for an automation should be planning for its testing by laying out the automated test cases you'll need.

So what is an "automated test case?"

It's a test of a specific portion of a workflow that can be automatically executed with no manual intervention required. These automated tests are performed by test robots, which can understand if there's a problem in executing the workflow and can document any issues while alerting development and QA teams.

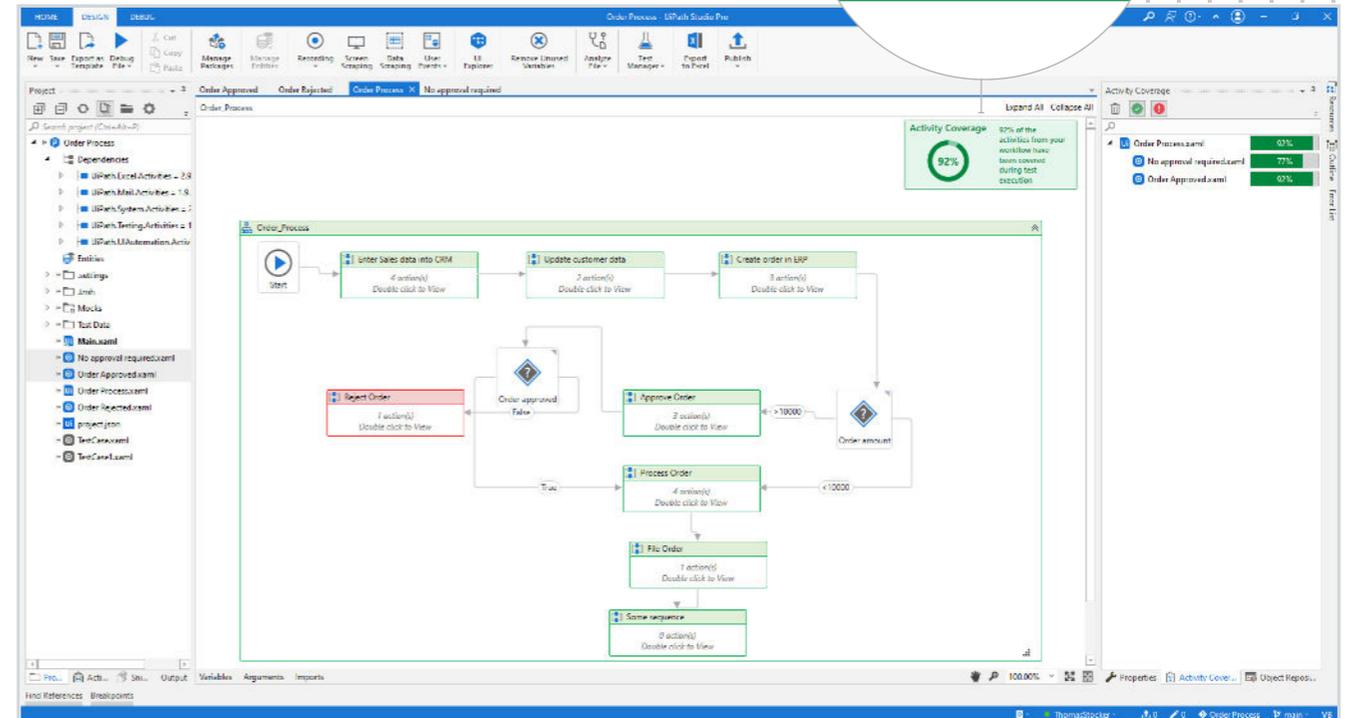
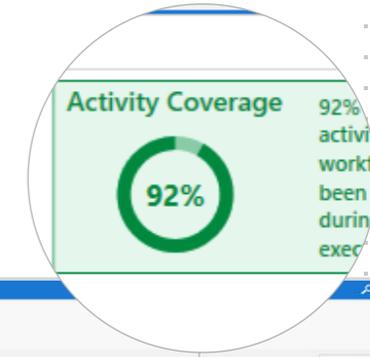
How do you build a test case? Currently,

most automated testing platforms require that you build tests by writing code-like scripts. But that doesn't have to be the case. For instance, if you use the UiPath Test Suite, it's not too much different from building any other automation. You'll build test cases in Studio using the same drag-and-drop interface and pre-built skills and activities you'd use to build regular automations. And you'll publish them the same way, in Orchestrator. These test cases will be stored within the workflow they're meant to test, where they can be accessed and deployed at any time. (With the UiPath platform, they can also be shared and re-used).

As you're planning, remember this: it's important to add enough test cases to test literally every activity. This is called "activity test coverage." After you execute tests, your testing platform should make it easy for you to quickly ascertain which activities were tested and what percent of your workflows were covered by the tests that were run. For example, the UiPath Test Suite provides this dashboard:



Example: UiPath Test Suite activity coverage dashboard



Continuously monitor every robot's health

Use test robots instead of people to test more processes, more often

The operating robots that execute your automations are working in environments where underlying applications, systems, and runtime environments are constantly changing. And some of those changes can disrupt a robot's flow. That's why continuous testing is particularly critical for RPA.

Luckily, an automated testing capability makes continual monitoring easy, scalable, and cost-effective. **Here's how it works.**

1. Define test sets

Test sets are groupings of related test cases that, for example, may focus on different parts of the same automated process. For instance, you may have processes for onboarding and offboarding that utilize the same applications. You might group those into a test set called "Employee Management."

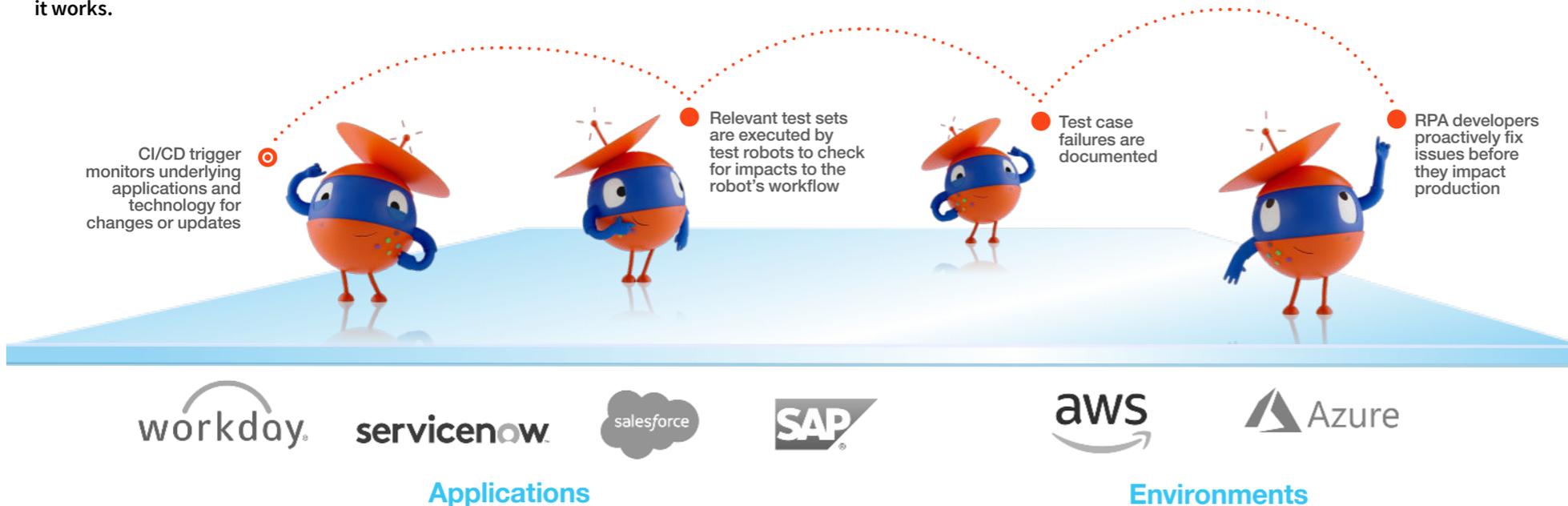
2. Orchestrate execution

Test sets can be executed manually or on-demand, set up on a specific schedule, or triggered when a continuous integration tool like Jenkins or Azure DevOps detects something new in a dependent application (see diagram below).

3. Monitor automations

When a test set is executed, a test robot does the work. Test robots make up your digital test team (and don't require much sleep). They are always available and when they find a defect, a robot will alert you and take a screenshot, so you can quickly triage and fix the problem.

This approach allows you to spot and fix specific issues before they impact operations. It can dramatically reduce maintenance time (especially useful for emergency break fixes!).



Did you know?

In two recent surveys, UiPath found that robot maintenance is the biggest challenge for our customers—and the task that RPA developers like the least.



Strive for an integrated and collaborative testing environment

You'll cut down on maintenance, drive reuse, and boost testing efficiency if you and your enterprise testing team use the same platform

To be more confident that your robots will keep working, it is important to coordinate with your application testing teams to test both the robots and the applications they rely on. That way, both teams can spot integration and coordination issues before they become big problems.

provide, allowing them to double their test coverage and accelerate release cycles. So it's worth taking a joint look to see if it makes sense for everyone to jump on board.

While you could use different testing platforms and still coordinate efforts, you'll find it's much easier to use the same platform. Just be sure that it can handle both RPA testing and the more extensive application and environment testing activities of the application team.

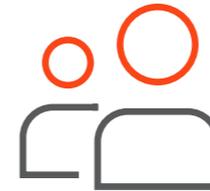
While you're exploring options, remember that the UiPath Test Suite can handle testing across the board. What's more, it gives application testers production-grade automation and orchestration that their current automated testing platforms may not

Did you know?

A top 5 US healthcare provider more than doubled its automation test coverage and shaved 2 months off their release cycles using UiPath for Application Testing.

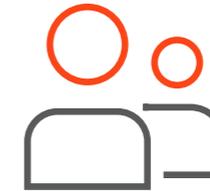


Benefits of a common test platform:
tightening collaboration between test teams and RPA teams; sharing and reusing components



Test teams

Evaluate software quality



RPA teams

Automate business processes

CREATE AUTOMATED TESTS

CREATE RPA ROBOTS



COLLABORATE

Test Manager

- Requirements
- Test Cases
- Test Sets
- Defects



AUTOMATE

Studio Pro

- Workflows
- Test Cases
- Activities
- Objects



EXECUTE

Test Robots

- Results
- Schedules
- Triggers

UiPath Test Suite: build more, test better, continuously monitor

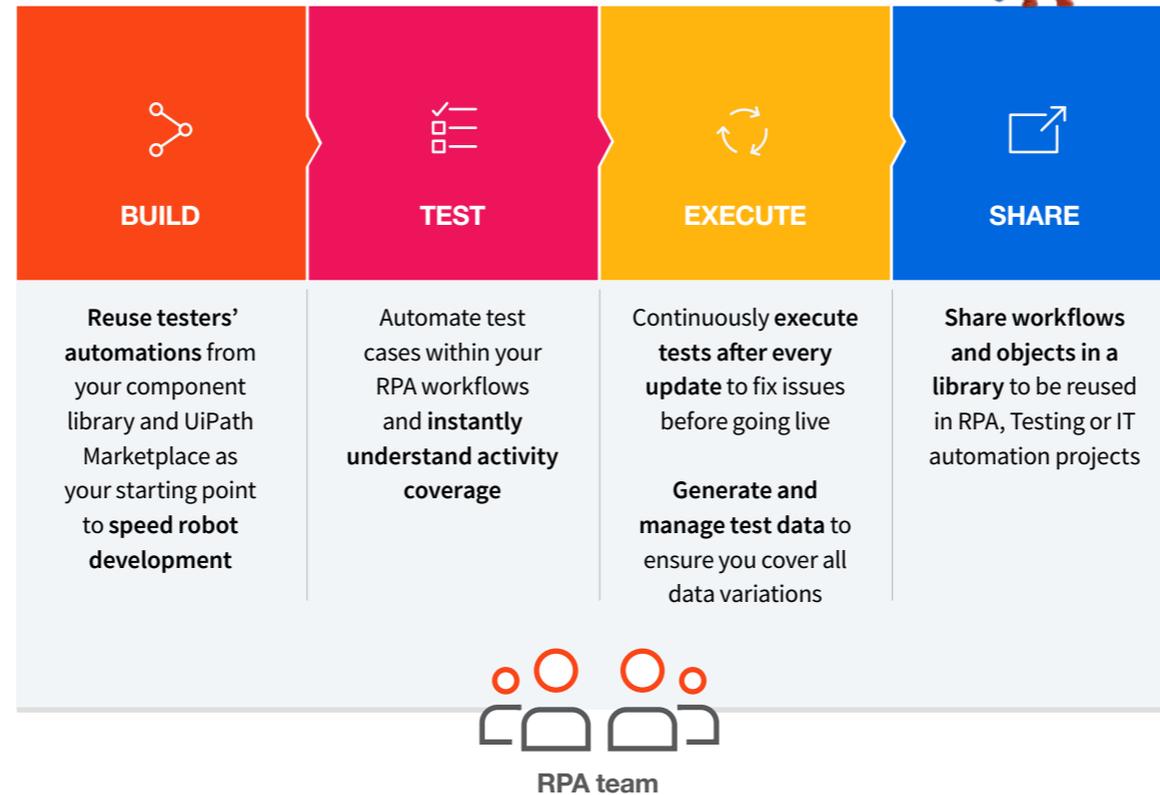
Make continuous testing continuously simple and powerful

At UiPath, we want our platform to enable every company to be a fully automated enterprise™. That means providing all the capabilities that allow companies to scale automation and take full advantage of it throughout their operations.

As we've discussed earlier in this paper, a core component of scaling lies in evolving from the waterfall development process and adopting continuous automation development. And one of the key capabilities to enable continuous automation development is continuous testing of automations from initial development through to ongoing operations.

In UiPath Test Suite, we've built an industry-leading capability to support continuous automated testing, monitoring, and maintenance across the automation lifecycle. We're the first RPA platform to offer automated RPA testing, and we continue to be the innovation leaders in providing an easy-to-use, scalable, flexible automated testing tool that provides superior testing capability to RPA teams as well as application testing teams.

How UiPath Test Suite make continuous testing and monitoring possible



We have a lot more to tell you on this subject:

If you're interested in having a conversation, please contact us.

[CONTACT US](#)

And if you'd like to explore more on your own, we recommend the following:

LEARN
Watch the Reclaim Your Weekend video

[WATCH THE VIDEO](#)

DISCOVER
Read the Test Suite datasheet

[READ THE DATASHEET](#)

TRY
Download UiPath Studio Pro for free

[DOWNLOAD UIPATH STUDIO](#)