

David Chappell

Understanding RPA Total Cost of Ownership

The Blue Prism Example

Sponsored by Blue Prism

Copyright © 2018 Chappell & Associates



DavidChappell
& Associates

Contents

- Defining RPA TCO..... 3**
 - The Components of RPA TCO3
 - RPA Styles: Comparing Enterprise RPA and Robotic Desktop Automation3
- Examining the Components of RPA TCO..... 4**
 - Purchasing Software4
 - Creating Initial Processes5
 - Creating Later Processes.....6
 - Executing Processes7
 - Managing and Scaling Processes7
 - Securing and Auditing Processes8
- Conclusion 11**

Defining RPA TCO

Robotic process automation (RPA) can be an effective way to automate business processes. But since a primary goal of process automation is to lower costs, anybody looking at RPA needs to think hard about the total cost of ownership (TCO) of the RPA solutions they're considering. These products differ in various ways, however, which can make it hard to compare their TCO.

This article walks through the elements of RPA TCO. We'll use Blue Prism as a concrete example throughout, but we'll also compare Blue Prism's TCO with that of another approach to RPA: Robotic Desktop Automation (RDA). The goal is first to provide a model for thinking about RPA TCO, then explain how and why Blue Prism TCO differs from the TCO of using RDA products.

The Components of RPA TCO

How should you think about the total cost of ownership for RPA products? One approach is to divide your total cost into six distinct components. Figure 1 illustrates this idea.

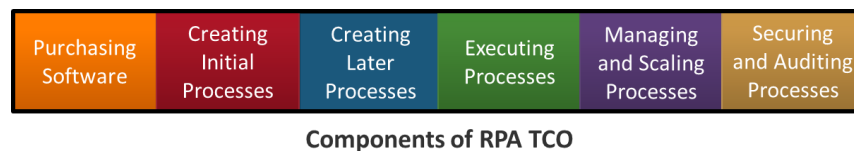


Figure 1: Determining RPA's total cost of ownership requires considering several different things.

The six components are the following:

- ❑ The cost of purchasing your RPA software.
- ❑ The cost of creating your initial automated processes.
- ❑ The cost of creating later automated processes. It's worth breaking this out separately because you might spend more or less time creating these later processes than you did creating your initial processes.
- ❑ The cost of executing your processes, i.e., what you spend actually running them on desktops or servers.
- ❑ The cost of managing and scaling your processes.
- ❑ The cost of securing and auditing your processes. While these costs can be hard to quantify, it's clear that weaknesses in these areas can cost your organization a great deal.

RPA Styles: Comparing Enterprise RPA and Robotic Desktop Automation

Before looking at RPA TCO, it's important to distinguish between two quite different styles of RPA: enterprise RPA, which is the approach Blue Prism takes, and Remote Desktop Automation. Figure 2 gives a big-picture view of how these two options address the fundamental problem of automating business processes.

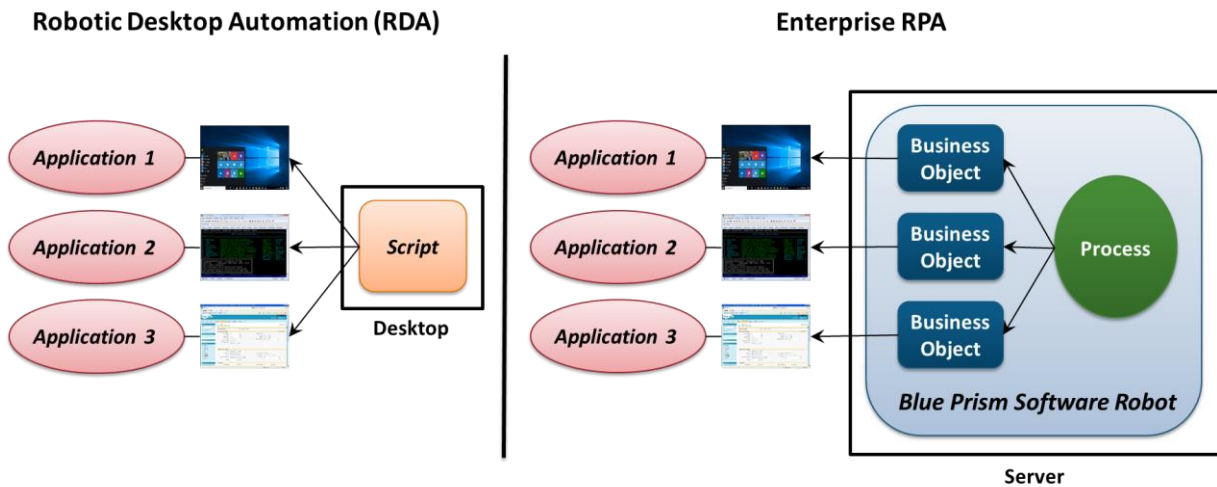


Figure 2: While RDA uses scripts, enterprise RPA solutions, such as Blue Prism, use software robots to automate business processes.

As the figure shows, RDA products typically create scripts that run on desktop computers, with each script implementing the steps in some automated process. Enterprise RPA, by contrast, provides software robots designed to run on servers. For example, rather than using a script, each Blue Prism robot implements automated logic using two components:

- One or more *business objects*, each of which handles interactions with a particular application or service.
- A *process* that carries out the steps in the automation logic. Every process relies on some number of business objects to handle interaction with applications and other services.

Even though both RDA and enterprise RPA fall under the umbrella of robotic process automation, the two approaches have different goals. While RDA offerings are largely focused on creating scripts for people to run on their desktops, enterprise RPA solutions such as Blue Prism aim at creating a centrally managed, scalable, and secure digital workforce using software robots. Both approaches have value, but as you might expect, the architectural differences between RDA and enterprise RPA solutions lead to differences in TCO.

Examining the Components of RPA TCO

Each component of RPA TCO plays a role in the overall cost your organization will incur in using this technology. What follows looks at these components, giving a sense of Blue Prism’s costs in each area and of how this cost compares with RDA products.

Purchasing Software

How RPA vendors price their software varies quite a bit. Blue Prism, for example, charges per software robot, while other providers use per-server licensing, per-transaction pricing, and other approaches. These variations make it hard to compare the pricing component of RPA TCO across vendor offerings.

One point to keep in mind, however, is that as Figure 1 suggests, the cost of RPA software probably isn't anywhere close to the majority of RPA TCO. If your organization purchases an RPA product but never uses it, then the cost of the software will likely equal your TCO. But as you automate more processes, the money you spend on the software itself will shrink as a percentage of TCO. The cost of the software is important, but it's almost certainly not the main factor in your RPA TCO.

Creating Initial Processes

Once you've purchased and installed your RPA software, you'll next use this software to automate your first business process. The money you spend on the people who do this—whether they're outside consultants or your own staff—is the next component in RPA TCO. An enterprise RPA product, such as Blue Prism, addresses this quite differently from RDA offerings, as Figure 3 shows.

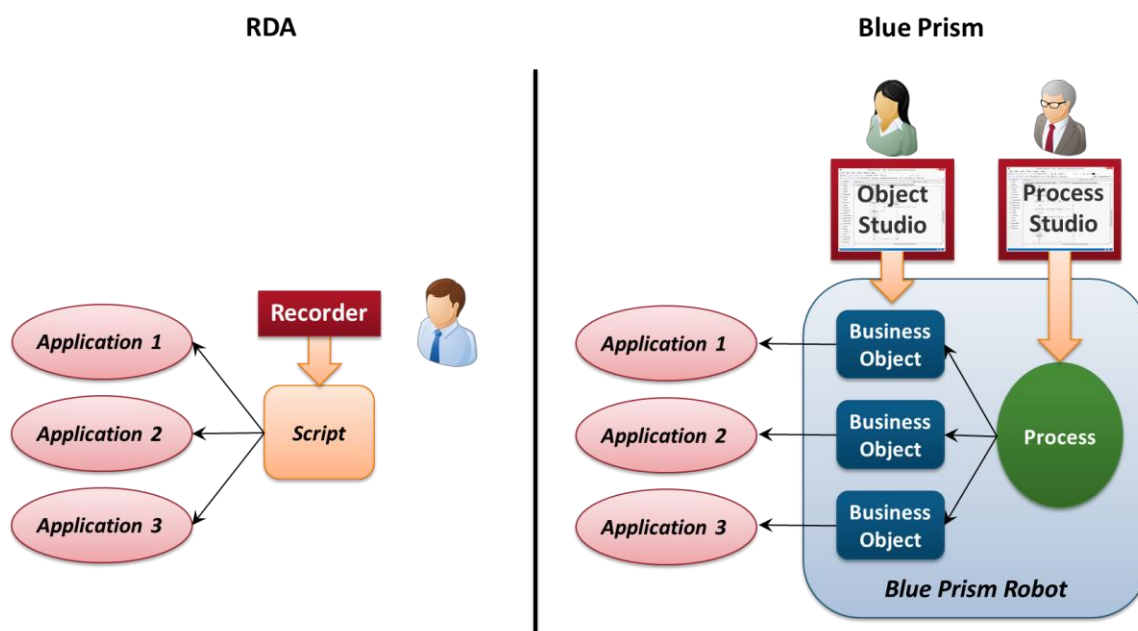


Figure 3: Creating new automated processes is different in RDA and Blue Prism.

In an RDA product, the typical approach is to have an experienced user start the product's recorder software, then carry out the process manually. The recorder remembers each step in the process, then creates a script listing those steps. This script can be executed over and over to execute the same process automatically.

Blue Prism doesn't rely on a recorder. Instead, IT and/or business people use a graphical tool called Object Studio to create whatever business objects are needed. These objects handle the low-level details of interacting with application user interfaces. Once these are ready, IT and/or business people use another graphical tool called Process Studio to define the steps in the process. When both are done, the resulting combination of process and business objects is assigned to a Blue Prism robot and executed.

As is probably obvious, using a recorder is generally faster than the Blue Prism approach when you're first creating automated processes. Since creating these initial processes takes less time with RDA offerings, this component of

RPA TCO is typically lower with RDA offerings than with Blue Prism. Still, Blue Prism chose this approach for a reason: it makes creating and modifying your automated processes easier over time, as described next.

Creating Later Processes

Assuming your first attempts at process automation go well, you'll next automate more business processes. With RDA offerings, this means creating more scripts. With Blue Prism, you'll create more processes and (perhaps) more business objects.

This is where the value of Blue Prism's object-based approach becomes clear. Rather than walking through every detail of the recording process again, as in a script-based approach, Blue Prism lets new processes reuse your existing business objects. This reuse can make it faster to create those processes. In fact, the more you use Blue Prism, the larger your library of reusable business objects will become. Over time, the cost of creating processes will go down—there's less work to do for each one—something that's typically not true of script-based solutions. Blue Prism also provides a standard library containing a number of pre-built business objects, including objects to access CRM applications, cloud services, and more.

This object-based approach also lowers the difficulty—and thus the cost—of changing existing processes. For example, suppose a new release of some business application changes an aspect of the application's user interface (UI). With a script-based solution, you'll likely face the challenge of finding all scripts that use this application, then modifying each one to reflect the UI change. With Blue Prism, however, you'll need to change only the business object that implements your interface to this application. Once you've done this, the updated business object is automatically installed into every process that uses it, as Figure 4 shows.

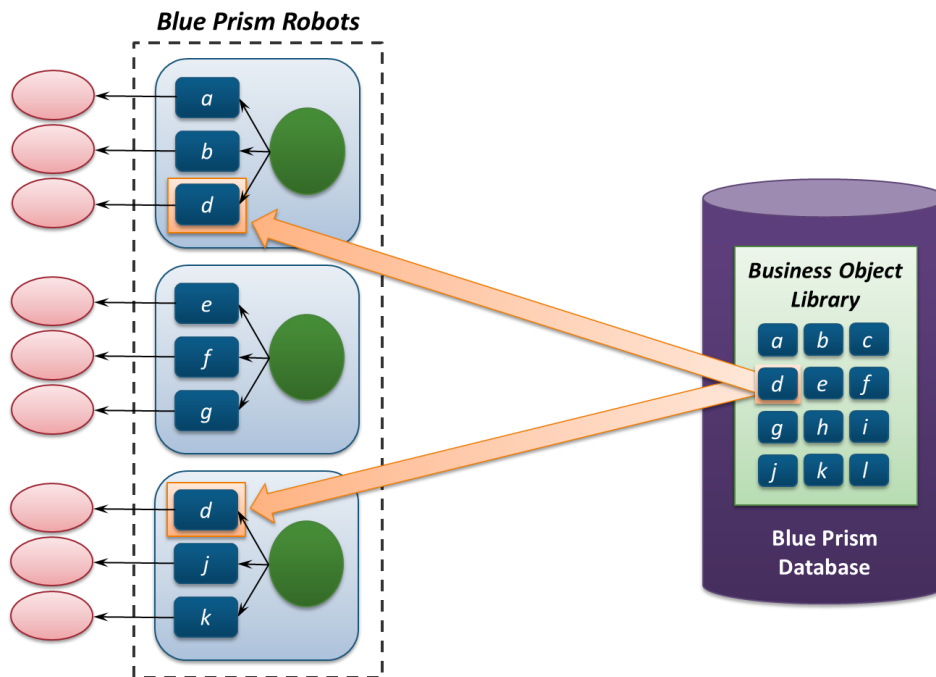


Figure 4: Blue Prism automatically propagates changes in a business object to all processes that use that object.

As the figure shows, your library of business objects is kept in a centralized Blue Prism database. In this example, object *d* is updated, perhaps to handle a UI change in some application. Once this update has been saved to the library, Blue Prism automatically finds all processes, running or not, that rely on this object, then updates them to use the new version. This approach both improves accuracy, since the change is made only once rather than to a number of different scripts, and lowers costs, since there's less work required to make the change. As a result, Blue Prism's object-based approach helps you lower this component of RPA TCO.

Executing Processes

The goal of RPA isn't to create scripts or objects and processes; it's to execute them. How much you spend on this TCO component is heavily dependent on how efficiently your automations run. The more efficient each one is, the less you'll spend on hardware to run them.

Even small differences in performance can add up. For example, suppose one of your automated business processes handles 100,000 transactions a month—not a huge number—and one RPA offering executes the process two seconds faster than another. This small difference adds up to 200,000 seconds per month, more than 55 hours. Because minimizing RPA TCO requires minimizing process execution time, taking more time up front to create faster automations is typically the right trade-off to make.

An important thing to understand about the performance of your automated processes is the difference in how RDA products and Blue Prism handle wait time. Recall that the scripts used in RDA are typically created by recording a user walking through the process. But since the process is interacting with applications, how long should the script wait for each screen to appear? Since the load on each application varies, this delay might vary as well. Some RDA offerings address this by inserting a fixed wait stage each time an application loads a new screen: two seconds, four seconds, or whatever seems appropriate. This simple approach can work, but it also slows the script down. What if a screen usually loads in only one second, but sometimes takes five? A fixed wait stage would need to allow for the slowest case, forcing the process to execute more slowly than it could.

Blue Prism addresses this problem by using an intelligent wait stage. Rather than waiting for a fixed amount of time, this wait stage can detect when a new screen is loaded. The result is faster process execution, which lets you use fewer robots to carry out the work. Performance improvements like this are an important part of lowering your RPA TCO.

Other aspects of Blue Prism also help keep execution costs down. For example, Blue Prism robots can communicate with one another, letting them avoid doing needless work. Suppose, for instance, that a CRM system used by many Blue Prism robots is down. The first robot to determine this can inform all of the other robots that the CRM system isn't available, then open one support ticket. Without this, your organization would waste compute time as every robot discovered the outage for themselves. More important, you'd also waste support time (and annoy your support staff) by generating multiple tickets for a single problem.

Managing and Scaling Processes

As described earlier, RDA products typically run on desktops. Blue Prism, however, is designed to run on servers in datacenters—it's an enterprise RPA offering. Figure 5 shows how this looks.

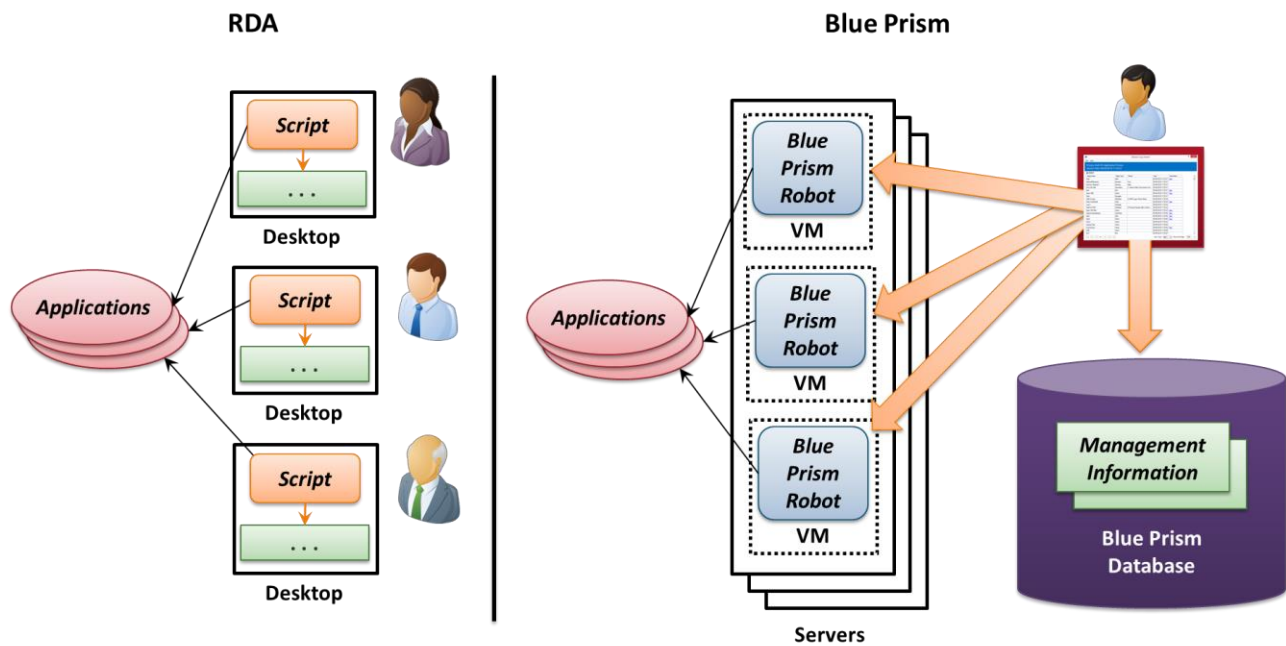


Figure 5: While RDA products typically run scripts on desktops, Blue Prism runs robots on servers in a datacenter.

The Blue Prism approach lets you manage your robots from a central management tool. It also stores management information about this digital workforce in the shared Blue Prism database, giving you one place to find what you need to monitor and control your RPA environment. Compared with managing multiple scripts spread across diverse desktop machines, this centralized approach can save time and money. Blue Prism’s goal is lights-out automation, with one or two people able to shepherd hundreds of robots. The result is a lower number for this important component to RPA TCO.

This approach to centralized robot management also makes scalability easier. With RDA products, scripts are often started and stopped by a user sitting at the machine on which they’re executing. If the goal is automating a process run by that user, this can work well. But if you’re trying to create a digital workforce, with robots that can be easily scaled up and down as needed, Blue Prism’s centralized approach is more effective. Rather than requiring multiple users to start and stop scripts individually, Blue Prism lets a single person start and stop robots as needed. The result is simpler, faster scaling, another aspect of lower costs.

Securing and Auditing Processes

When people execute business processes, especially mission-critical processes, most organizations have procedures in place to ensure security and to provide an audit trail of what those people do. Automating business processes with RPA shouldn’t change this; security and auditing remain fundamental requirements.

Perhaps the most basic requirement for achieving these things is a way to add authorized users to your RPA environment, then define what those users are allowed to do. RDA solutions typically do this differently from Blue Prism, which leads to a difference in TCO. Figure 6 illustrates this.

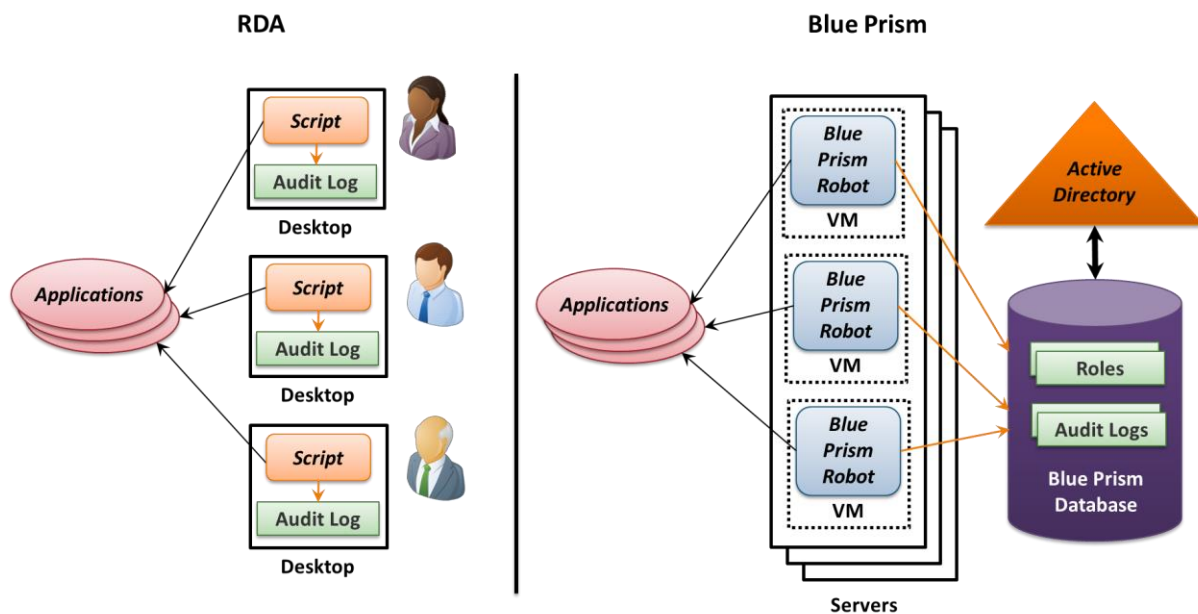


Figure 6: Unlike typical RDA offerings, Blue Prism stores role definitions, audit logs, and other information in a central database.

In an RDA product, users might login to their desktop machine, then run scripts under their own identity. Each user might also have the ability to change the script running on their machine. In Blue Prism, robots—running on servers in virtual machines—are assigned their own identity. Blue Prism can rely on Active Directory to do this, adding robots to the directory service just like people. Robots can then be assigned roles, letting them access only specific applications and data. Blue Prism also allows creating roles with different capabilities for people. For example, one user might be allowed to create new business objects and processes, while another can only run existing robots.

Why is this detailed level of security required? Because just as a user might find a way to subvert a process, e.g., by siphoning off money to their own account, so might the creator of a robot. Without strong security in place, a fleet of robots can cause significant problems in a very short time.

A related issue is auditing. Without a reliable audit trail, there's no way to prove who did what. As Figure 6 suggests, RDA products can sometimes offer an audit trail on each desktop, but the security and completeness of this data can vary. Blue Prism approaches this problem quite differently. In part because its original users were in the financial services industry, Blue Prism maintains a complete audit trail for every robot, every user, every administrative action, and more; every action is logged all the time. This information is created by Blue Prism itself, then stored in the Blue Prism database.

This detailed audit data is important for several reasons. First, it provides an effective way to track what's happening. If a robot is behaving incorrectly, for example, its audit trail will contain a fine-grained record of what the robot is doing, helping you find and fix the problem. More important, Blue Prism's audit trails support non-repudiation. Suppose one of your organization's employees has subverted a robot to behave in an inappropriate

way. The audit data will let you prove (in court, if need be) exactly what the robot did and which of your people was responsible for the subversion.

Determining the cost savings provided by strong security and non-repudiation can be an inexact science. Still, it's clear that Blue Prism's audit trail can help lower the cost if you're audited for compliance with HIPAA or other regulations, since you'll already have a reliable record of events. A reliable audit trail also increases the likelihood that criminals in your organization will be caught and convicted, a deterrent to bad behavior.

More generally, it's clear that security problems can cost organizations large amounts of money, including declines in their share price. In some industries, they can even result in jail time for executives. While the savings can be hard to quantify, no one can debate the importance—and the potential cost savings—of strong security and auditing in a RPA environment.

Why Hackathons Aren't the Best Way to Choose an RPA Product

One approach to choosing among various RPA products is to hold a hackathon. The idea is simple: Get some of the people who will use the product together in a room for a day or two, then see how easy it is for them to create business value. What's not to like?

The answer is that hackathons measure only a small part of an RPA product's value: how easy it is to create relatively simple automations. If your organization is looking for a way to create scripts that users can run on their desktops, this might be your main criteria for the decision. If you're looking to create a digital workforce, however, with the lowest overall TCO, a hackathon isn't going to help you make a good decision. Because hackathons focus on just a small part of an RPA product's functionality—creating initial processes—and they don't help at all in understanding each product's overall TCO. Appealing as a hackathon might be, don't expect it to help much in choosing the right RPA offering for your organization.

If you're looking to use enterprise RPA to build a digital workforce, doing a proof of concept (POC) is likely to be more useful. Your POC should let you implement an actual process in your own environment, then see it in action. This might take a month or two, but the experience will give you a much more accurate sense of what's needed to create a manageable, scalable RPA environment.

Conclusion

The total cost of ownership for an RPA environment has several components. Predicting what your TCO will be means looking at all of them. While specific numbers will vary, Figure 7 shows the typical relationships between these components for Blue Prism and RDA offerings.

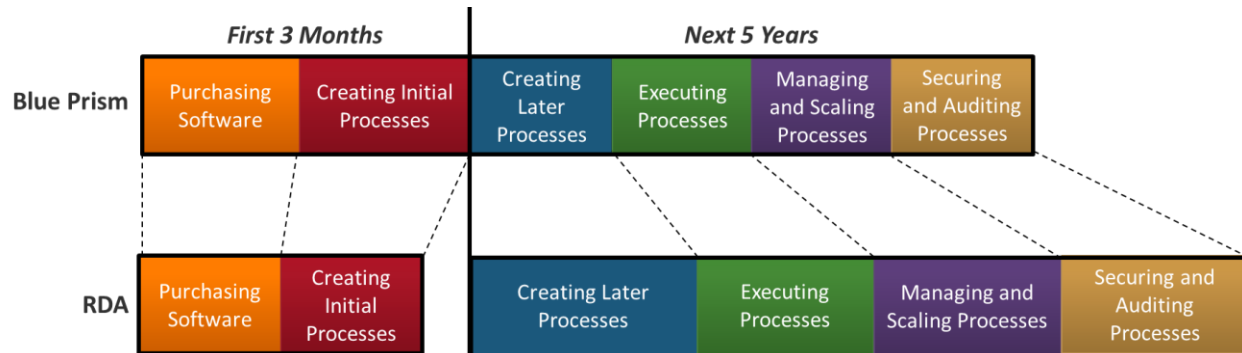


Figure 7: For creating a digital workforce of secure, well-managed robots, Blue Prism is likely to provide lower TCO than RDA.

As the figure shows, these two approaches vary in each component of TCO. To understand the differences, it's useful to look at two time periods: the first three months after buying the product and the next five years of using it. The first three months includes two TCO components:

- Purchasing software: Blue Prism isn't the least expensive RPA product on the market today, so it might well cost more than an RDA offering.
- Creating initial processes: Blue Prism's object-based approach takes more time than recording a user executing a process, which translates into higher cost for Blue Prism.

Looking at just these two TCO components, Blue Prism likely costs more. These higher costs occur only at the beginning, however. Blue Prism's overall cost advantage stems from differences in the TCO components that determine your costs over a five-year lifecycle of using the product. These components are the following:

- Creating later processes: Because of object reuse, creating processes with Blue Prism gets easier—and thus less expensive—over time. And because changes to a business object are automatically propagated to all processes that use that object, updating processes is also less work with Blue Prism.
- Executing processes: Blue Prism processes typically run faster than RDA scripts, in part because of things such as Blue Prism's intelligent wait state. This lowers costs because it requires less hardware to accomplish your goals.
- Managing and scaling processes: Because Blue Prism is designed to be centrally managed and scaled, both of these things are simpler—and thus cheaper—than with desktop-oriented RDA products.

- Securing and auditing processes: In part due to its early roots in the financial services industry, Blue Prism has stronger built-in security and auditing technology than typical RDA offerings. As many organizations have learned, the cost of ineffective security can be enormous.

If you're looking for a fast and simple way to create desktop scripts, choosing an RDA offering might well be right for you, despite its higher long-term TCO. But if you're looking to create a scalable, secure, server-based digital workforce, expect Blue Prism to provide lower TCO over the long run.

David Chappell is Principal of Chappell & Associates (<http://www.davidchappell.com>). Through his speaking, writing, and consulting, he helps people around the world understand, use, and make better decisions about new technologies.