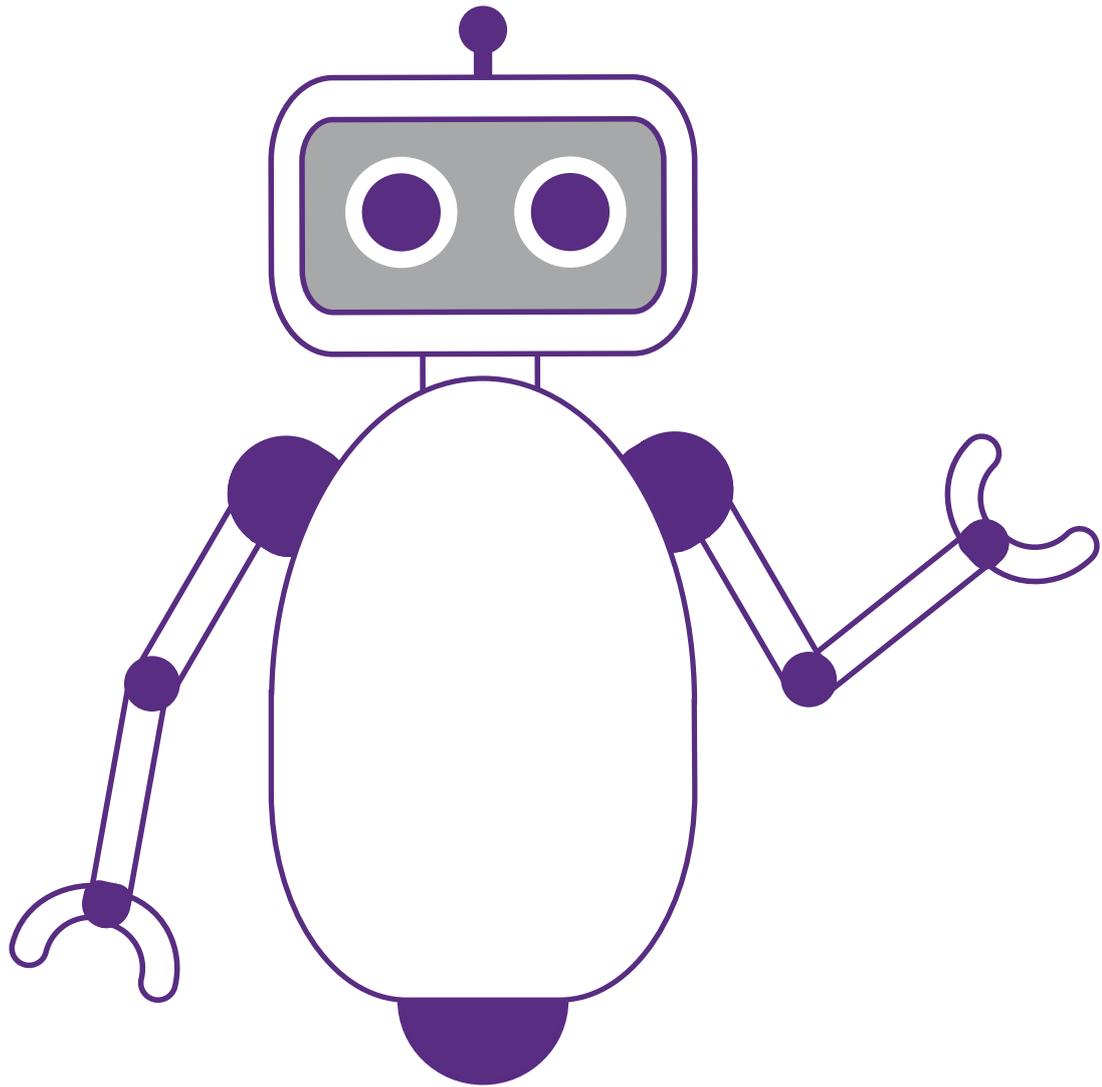


An Introduction To  
**Robotic Process  
Automation**

---



## An Introduction to Robotic Process Automation (RPA)

Robotic Process Automation has made headlines recently, seems to have come of age and is undoubtedly here to stay. There is, however, confusion as to exactly what RPA is, what it is good for, and what it is not. This eBook aims to provide; a background for those interested in an introduction to this topic, identification of the benefits that RPA can provide, the different types of implementation and some of the lessons that we have learned over the last couple of decades working with these technologies.

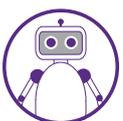
This eBook forms part of a series which includes: 'The Little Book of RPA', highlighting a wide-ranging collection of examples of RPA in action, and 'RPA Implementation Guide: Delivering a Successful RPA Project', which suggests some dos and don'ts of successful implementation based on our many years' experience in this field.

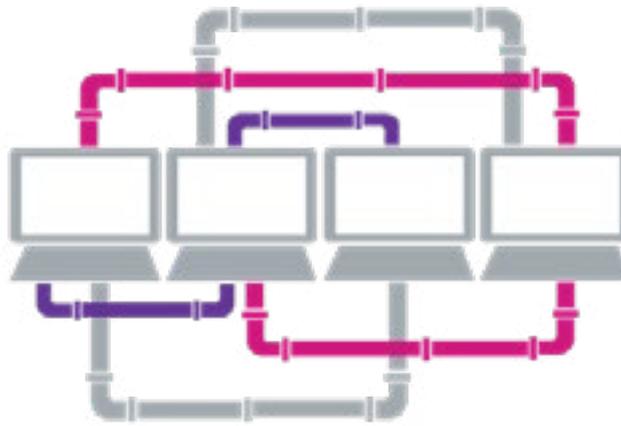
The benefits of RPA and 'Digital Labour' are massive and transformative, however, it does need approaching with realistic expectations, thought and planning.

## A Brief History of RPA

What surprises many is that RPA is far from new. It is an evolution of technologies from over thirty years ago, firmly rooted in the mainframe era. Just as the basic non-portable telephone has morphed into today's smartphone, RPA technology has evolved to a similar extent, covering just as much distance. Today we can deliver on a breadth and at a scale unimaginable when the technology was first introduced.

Back in the eighties, mainframes and mini computers were at an organisation's heart. PC's were used to connect to them and brought productivity tools such as spreadsheets and word processing into common use. 'Screen scraping' technology was invented to deliver automated data extraction, integration and upload in the PC world and to save time automating mainframe tasks. Over time this became increasingly sophisticated and involved both 'on-screen' visible activities and 'lights out' pure server variants. All of this was based upon the coding of a, well-defined, repetitive task and having the machine follow the same process that a human operator would follow. – Simple!





Leading up to the year 2000 and the 'millennium bug' kerfuffle, most mainframe and mini systems were decommissioned in favour of 'modern' client server, Windows and browser-based systems. To some extent, these legacy methods were forgotten or went underground. It soon became apparent, however, that the new systems are also plagued by the very same issues as the legacy systems. They too lack integration and suffer from the need for repetitive tasks, often involving system to system re-keying, or the repetition of processes with only small variation. For example, the production of month-end departmental reports, or matching a CRM or eForm-booked process with a back-office system entry.

Time and technology move on. The level of interaction with the systems targeted by these tools has grown significantly more sophisticated. The environments, scalability and manageability have all increased proportionately. The comparison between 'Bakelite' telephony and the modern Smartphone is apt. This new level of interaction deserves its 'New' title as it is a million miles away from the legacy of 'screen scraping', albeit we should not dismiss its origins and the lessons learnt over the decades of implementing these solutions. RPA has well and truly arrived.

This space is being somewhat confused however by the hype-cycle associated with Artificial Intelligence (AI). Machine Learning has a contribution to make in this space, but we are still a long way off it being able to deliver properly, and conversations are conflating Robotic Process Automation with AI, Chatbots and the rest. At the time of writing, reliable AI systems capable of examining unstructured data such as blocks of text, automatically diagnosing and prescribing appropriate action are still a very long way from being production-ready.

What we see more commonly is the employment of 'Chat Bots' with a limited vocabulary who may, in turn, request a non-intelligent Bot to look up an item such as an outstanding balance. These very different technologies tend to be 'lumped together' in a bid for companies to be seen to be 'at the forefront of technology'. At this point, we would advise they are approached separately and with some caution.



## Main Types of RPA

Before examining its usages and the various benefits of RPA, it is worth recognising the two main methods of deployment. These are often referred to as 'Attended' and 'Unattended' implementations and are described in more detail below:

### Attended

This is where a robotic software agent is placed on the desktop of a real person. Typically deployed in a call centre environment, the process(es) that this 'Bot' performs happen in front of the agent, taking control of their environment and saving them time. A good example might be recording a change of circumstances or looking up multiple balances across different systems and laying them out for the agent to have a common view.

These processes can be triggered by the injection of a button or 'Control' into the main application or the addition of some independent UI (User Interface) on the desktop. Alternatively, some RPA products allow for the insertion of 'triggers and traps' to be created on top of existing applications. Effectively invisible to the user, these will perform an action based on the use of a specific bit of UI. For example, if details have been changed about some circumstances, then when the 'submit' button is pressed, these might be transmitted or stored before proceeding, but without the user being aware of this.

Attended RPA offers the user direct and visible evidence of the system performing tasks for them, providing direct feedback about the time they are saving. This can quickly become tiresome, however, as it can be seen to be preventing them 'getting on' with the next task.

### Unattended

This is where the 'Bots' run independently in some form of virtual environment, on a server or in the cloud. The Bots in question are purposed against specific applications, processes or transactions, often having processes sub-divided between them like in a production line. This is a highly controlled environment and as it is virtualised, Bots can be cloned simply and quickly to achieve significant scalability. For example, in one instance we used a team of 45 Bots to migrate over 1.5 million documents from one system to another in very short order.

A key capability to support this is that of the 'Bot Manager' which can allocate, supervise and audit individual Bot performance, recording and reporting any issues to administrators. (Just as there would be a manager or supervisor for a human team). This is usually supplemented by some form of transaction orchestration and queueing.

Aside from scalability, the great advantage of this approach is that they can run 24/7, are in a highly secure and stable environment and they don't stop or get bored. Unlike the Attended model, this method does not impede the work of a human. Furthermore, they can be trusted with highly sensitive information as they don't gossip.



## Main 'use cases' and Benefits of RPA

There is no doubt that properly deployed RPA technologies can save a huge amount of manpower and bring a level of efficiency and agility to an organisation, supporting business transformation programmes. We see the main benefits falling into three main categories:



### Scalability

RPA brings the ability for an organisation to scale up its operations and transaction volumes without having to increase its human workforce. In times of austerity it is often used to cover gaps left by shrinking budgets. This scaling can be achieved in minutes rather than months; significantly less time than it takes to recruit and train a human team.



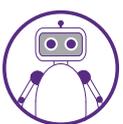
### Quality

Once implemented properly and given accurate data to work from, a Bot is 100% accurate, delivering perfect quality all the time. Bots are not prone to transcription errors or transposing one set of digits for another. They are also forced to obey the business logic of any target application, so they don't interfere with the databases of a vendor. Also, they can often provide a more comprehensive audit trail than the target systems that they drive.

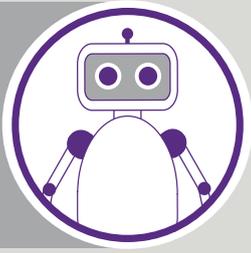


### Agility

The other vital part of the mix. Having a pool of 'Digital Labour' on tap means that an organisation can create new processes very quickly, in days and weeks rather than months, taking advantage of existing systems rather than having to acquire new ones and additional staff. Once the need has passed, transactions can be taken out of service just as simply, without a rump of unused new system investment or workforce redeployment to worry about.



Turning then to the question of practical 'use cases', we typically see RPA deployed in the following circumstances:



## Process Automation

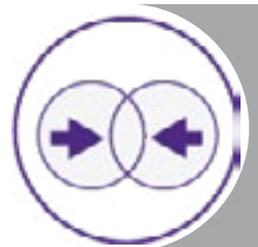
Simple though this may seem, there are many people in many organisations who spend their time re-keying the same data between multiple systems. This is not efficient. This has been shown to introduce errors and serious bottlenecks into a business process. Our experience has shown that in most enterprises there is a surprisingly large amount of re-keying still happening.

In one organisation, Bots have been deployed to interrogate a wide range of different providers of underground services prior to them engaging on their own ground works. On average this means retrieving some 25 different maps from different systems in different formats from over 350 possible providers. The human would have to enter sixteen-digit coordinates over and over again, for each provider, for the same work site. Now a team of Bots does this for 38,000 jobs per month, retrieving and reformatting some 12 million maps a year.

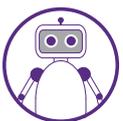
In an 'Attended' example the Bots help CRM operators by automatically fetching copies of letters from a document management system based on the CRM record in view. This saves the operator time and allows them to deliver a better customer experience.

## Data Integration

The reason that re-keying is often needed at all is a lack of appropriate and useable Application Programming Interfaces (APIs). Even when available, these are often complex, require advanced programming skills, can be very expensive and may only cover core functionality rather than the full system or any subsequent customisation.



The RPA platform can provide a surrogate interface for the back-office system and use its Digital Labour 'at scale' to provide the required transactions. This may be viewed by some as an inefficient way of achieving the desired result, but the Bots can move surprisingly fast, outstripping human operators and, in some cases, can be even faster than vendor API calls. At least 50% of our customers use RPA for this purpose. (For further information on this topic, see the Buyers Guide to APIs, available on the NDL website [here](#)).



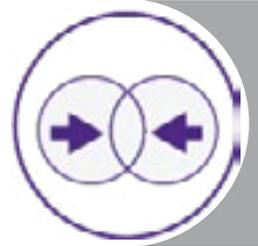


## Synchronisation

Here the Bots simply reflect a single version of the truth. If a change is made to one system such as a 'change of circumstances', then the Bots amend this across the other back-office systems in the organisation. This can happen on the operator's desktop or simply be carried out 'server side' as an unattended process. Sometimes a hybrid model is used to trap a human desktop driven change, but rather than slowing down the agent in the call centre, the desktop Bot transmits the change to a central 'co-Bot' and the remaining systems are checked and updated server side.

## Extraction

Quite often data can be hard to get at and may require access to sensitive records that are above the clearance level of an individual. For example, if we have a medical data set and we wish to extract the number of prescriptions for a specific drug, and the postcode of the GP surgery, to build a map of the prevalence of a condition.



The software robots may scan each patient record, note the date and dose of the medications in question and return the postcode of the GP practice concerned. No human has had to have access to the confidential patient record, but we can make anonymous data available to an epidemiologist who may be able to design a health education campaign with this information. – As a result, RPA technologies have a fantastic role to play in 'Open Data'.



## Migration

In its simplest form, 'lift and shift', copying large amounts of data from system A and either placing it into a known format for import or entering it directly into system B. The advantage here is that sometimes metadata and data from other cross-referenced sources can also be added to the data set. It is not uncommon for projects to employ teams of robots to move millions of records in a short space of time - a process where accuracy really matters.

We have examples of many other less common 'use cases' such as 'attribution look-up', where a Bot simply answers yes or no to a requester, preventing human access to a sensitive record such as in a medical, financial or benefits setting. An example being: Does this family receive benefits? If so, their children can receive a free school meal (passed to another Bot to add them, or to change a setting). If they receive free meals, they are entitled to a Uniform Grant (another setting, or a process delivered by another Bot). Even if the process is passed to a human, no access is needed to any details or financial information, a simple 'yes' or 'no' suffices. The Bot is totally trustworthy.



## What Can a Robot Do? How Does It Do It?

The examples above are great but how are they achieved? In the simplest terms, a software Bot can do what you or I can. However, just as it can do many things more accurately and at speed, with no errors, there are some significant limitations which also need recognising.

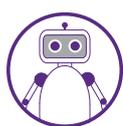
Essentially, the software Bot can be used to 'drive' desktop software the way a human would. This could be looking things up, entering data, capturing results - the whole range of software usage. All that happens is that we examine the way a person performs a process, codify it, identify any variables and then provide an environment in which that process can run.

There is a whole raft of technology supporting this, making sure it is secure, resilient, scalable etc but that is not important at this point. Although we can build quite sophisticated logic into these processes, it is important to understand that they are not intuitive in the same way a person is. To that extent, a software Bot can answer a balance query, copy and paste information or decide if a due date has expired, all day every day, but they are not intelligent.

It follows, therefore, that what a Bot will not do is read a page of narrative and pronounce a diagnosis, interpret Shakespeare or create a meaningful free-flowing dialogue. These things are the realms of people and possibly in the future of AI. There is some fantastic demomware out there and the press are getting very excited about the possibilities. Although it is coming, we are not yet at the point where we can truly trust these algorithms. Go for a ride in a 'self-driving' car if you require evidence of the current level of the art.

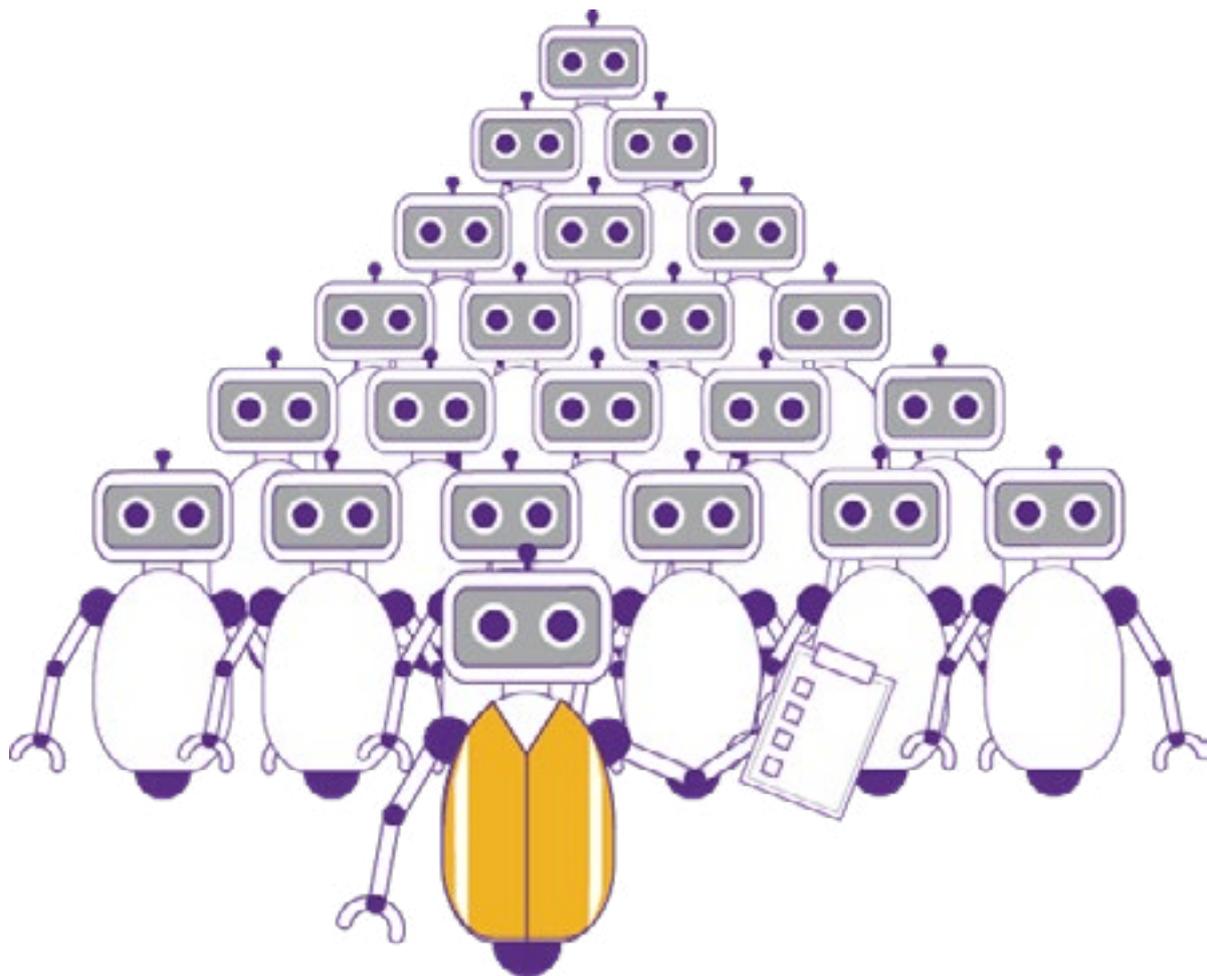
In addition, with AI, the level of Bot 'training' that must go into a single recognition process is vast and long winded. A recent project required the Bot to be trained on some 35,000 samples to achieve 90% accuracy on one recognition task; and only human review could highlight the 10%. Given that this is for a single process, you have to question the current level of scalability.

Machine learning is coming and at a pace, it is exciting and it holds huge possibilities, but it is still the preserve of the lab rather than the wild. We suspect that in 2025 we will still be waiting for our current expectations to be met.



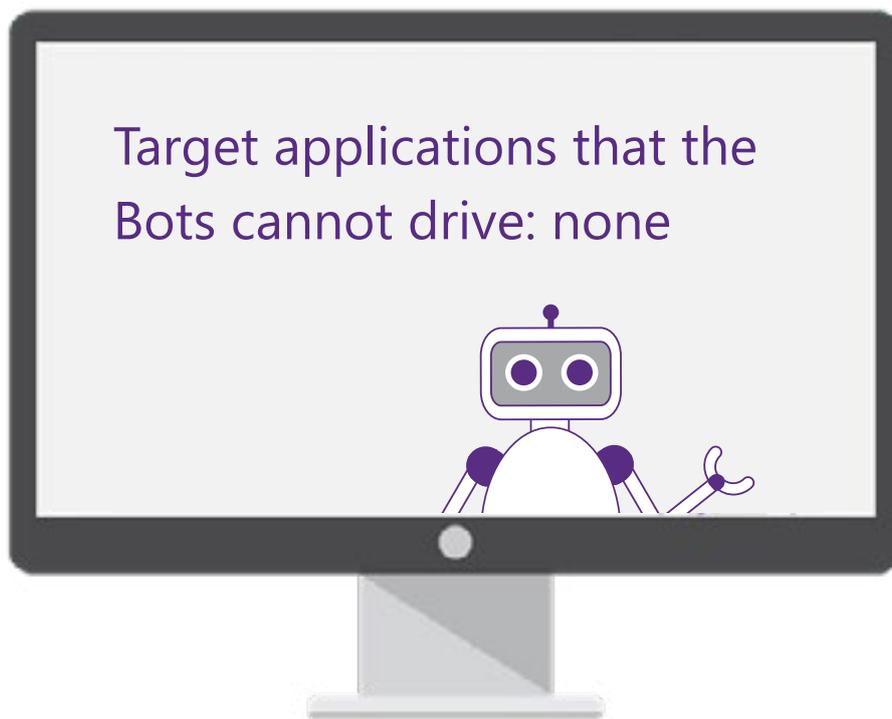
We make these points as the press and the industry are mistakenly combining these two areas accidentally, and sometimes deliberately, and will continue to 'hype' the technology. There is a danger of unrealistic expectations being set, especially with senior management on the receiving end of confidently delivered statements by vendors. We would advise extreme caution if betting your business processes on AI for a little while yet.

So where does that leave us? If you have a repetitive, well-structured process, with clear, logical and well-defined business rules, working on structured data, then RPA is perfect for you. RPA can save vast amounts of time, delivering accuracy and efficiency. If, however, you are working on intuitive, nuanced decisions and processes that require the understanding of unstructured information and balanced decision making, then RPA is not the answer. We propose finding the former and applying RPA to those, saving the time and money to allow the humans to really focus on the latter. After all, it's what we're good at.



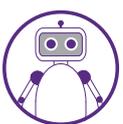
## So How Do They Do That?

Ignoring all the wider infrastructure requirements of an enterprise grade platform, simply understanding some aspects of how a Bot drives an individual system may help position the technology better. We have already established that the Bot drives the system the same way a person does, but what is really going on?



Humans see a system on a screen. On that system, there will be fields for data and labels explaining to the user what a field is. There will be buttons, drop-downs, tabs, text, pictures, a plethora of, what are known in the I.T. world as, 'controls'. We then have ways of interacting with these through a keyboard, the movement and clicking of a mouse, using short cuts like copy/paste to inject values into fields etc.

These systems may present themselves as desktop apps or be browser based and they could be created using a wide range of technologies such as traditional Windows desktop apps, Browser HTML, Java, Silverlight etc. What they all have in common is that they interact with users, and with the operating system (OS) upon which they work. In 99% of the cases in the business environment this is Microsoft Windows.



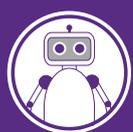
RPA technologies know how to interface not only with these 'controls' and input methods but also with the underlying operating system. This is important as it is this interaction with the OS message layer that delivers many of the speed and resilience features that make today's RPA tools so effective. If all else fails there are the good old standbys of replicating key stroke, mouse controls and even Optical Character Recognition (OCR) techniques that can be brought into play, but these are now rarely used, except in extremis.

One useful, but sometimes overlooked feature, is the ability to interact with pictures as well as data, capturing images or sensing colour changes on controls; for instance, a red green free/busy indicator on a back-office system.

What this all boils down to is that the technology being employed has come a long way since the early screen scraping tools, usually offering half a dozen different ways from a credible RPA offering to drive back-office systems. It is often just a case of picking the most appropriate tool or 'method' for any given target application from your RPA tool box.

Wrap up these Bots in nice secure virtual environments, add team management, security and audit and you have a very powerful agent for business transformation. Given all the different possible ways of interacting with back-office systems, it should be no surprise that we have yet to find a target application that we cannot drive, even in some of the highest security, locked down environments.

Much more information about how to deliver a successful RPA project, the benefits and pitfalls are included in our eBook 'RPA Implementation Guide', which can be found on the NDL website [here](#).



## Alternative Approaches to Implementing RPA

There seem to be two main ways that RPA offerings are created, and in turn, two main ways of implementing RPA projects. Taking the implementation choices first the alternatives are 1). Self-Serve, 2). Outsource to a consultancy, 3). A hybrid of options one and two.

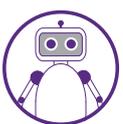
Power Tools	Higher Cost More Flexible	Lower Cost More Flexible
	Higher Cost Limited Reach	Lower Cost Limited Reach
No Code	Outsource	Self-Service

RPA is now a big business for the consultancy practices as it is closely aligned with business change programmes. They will advise on the whole business process transformation, specify what they believe needs to be done, then use the RPA tools to implement all the way through to 'Business as Usual' state. The scale of this varies dramatically with project budgets ranging from millions down to hundreds of thousands of pounds. They are not inexpensive which is why the large consulting groups now all seem to have teams dedicated to implementing RPA, but the software to enable this is usually a very small fraction of the total project cost.

The upside to using consulting firms is that the client gets the benefits of their experience, their business change advice and outsourcing the project risk. The advice and perspective on business transformation is especially valuable as there is little point automating a nonsensical or poor process. The secret of success with RPA, as with any project, is being clear about what you are trying to achieve and the project specification. No one should know your business better than you do but sometimes having an outside agency brings perspective and can act as an independent agent for change.

There are of course down sides. Cost is clearly the biggest, coupled with change management. Changes to requirements once a project starts are usually where the traps are, affecting both time and money. Given the complexity of some transformation programmes and the elapsed project time, it is almost inevitable that things will change in the business, so we would urge those planning to go down the external consultancy route to make sure they have a significant contingency.

So, what of Self Service? If you have the IT resource and a clear vision of what you want to achieve, then the nature of the tools available means that the level of expertise required is not actually that high. Indeed, in our experience, some 90% of customers usually take this route, sometimes starting out working directly with NDL or a partner for the first project in a hybrid model.



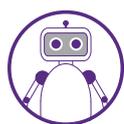
In these cases, we are usually responding to a specific project need, where a business problem has been identified. The business understands the challenge, knows what it wants to achieve and has some level of IT resource, either directly or contracted, available to deliver the project. Once the first small project has been delivered, this builds confidence and serves as an inspiration to other areas of the business. At which point more projects tend to be commissioned, each delivering an iterative saving.

By self-serving and starting with a tactical, agile approach, the savings from the first few projects free up resources to be applied to others, allowing a team to be created who become experts in the application and implementation of RPA. With these resources in place, not only can the business respond to process change, it can also reap the longer-term business agility benefits of RPA.

Turning then to the tools available to help deliver RPA. These seem to fall into two categories. First, there are those that claim to provide some form of 'no code' offering, using activity recording and 'drag and drop' techniques to define the process required to be automated. These are great for very simple processes, however as soon as you step outside the boundaries or wishes to introduce decision making logic, you can quickly run into barriers.

Second, at the other end of the spectrum, there are tools and add-ins for heavyweight developers, which plug into Visual Studio or other developer environments. These expose a set of APIs the developer can 'call' as part of their hand-developed project.

Here at NDL, we sit somewhere in the middle. We provide a graphical tool set that interacts with the target application(s) in a natural way and a studio where the process is developed. However, instead of having to write code by hand, selected actions just automatically generate the code for the person developing the script, 'under the hood' as it were. This means an IT literate person can develop scripts without being a heavyweight developer.



Any code generated is all industry standard, so any developer should be able to understand and modify it, if needed. This means that, if a process does require some special extension or business logic adding, it is a relatively simple matter for someone to write the code to extend it. For those who prefer a full developer environment, they get access to the controls as APIs and can develop in whatever way and language suits them. In addition to the development of a process, there are also some other considerations that we would suggest those looking at implementing RPA consider too. These are:



### Licencing

Is it per-robot, process, desktop or even charged by machine rather than human time? If you are considering a per-robot versus platform or process type licence, then it is worth considering how many you licence. It is usually more efficient to break a process down into parts, having different Bots delivering different bits of the overall process rather than being stuck with one going from end to end. (Think of the difference between a production line and a lone artisan).



### Management

What controls are offered for the deployment and management of the Bots? Can they be teamed, load-balanced, cloned, brought in and out of service? Is there sufficient information for your administrators to ensure that you can have a high quality of service 24/7? In addition, what level of Bot and transaction auditing is available?



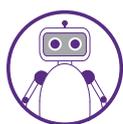
### Security

Aside from management and audit, can the Bots be secured and can any traffic between Bots and their controllers be strongly encrypted?



### Interfaces and triggers

How can a process be started? Is it Web Services, Database or folder driven? On the desktop, can you inject buttons into apps, provide new UI or remain dormant, responding only to in-app traps and triggers?





## Orchestration

Is there any? How are transactions managed, prioritised and recorded as completed? Can the results of process A be passed to process B or C depending on the values of those results?



## Environment

What virtual architectures are supported to allow scaling? Will it run in the cloud in environments such as Microsoft Azure or AWS?

The offerings available can be used to deliver some very complex processes that stand up to the rigours of any enterprise. We are a long way from the amateur macro running on a desktop in the corner. That said there is no reason why a very small and simple process, done well, can't make a big difference to an organisation's efficiency.

## Summary

Setting aside the 'snake oil' and conflation with AI, there is still the distinct whiff of the emperor's new clothes surrounding RPA. We would argue that RPA is merely an evolution and rebranding of some very well proven techniques and technologies. Technologies that we have been working with customers to implement since the beginning of the new millennium, if we ignore the experience brought forward from the mainframe era. The capability of RPA to deliver huge benefits and savings to an organisation is, however, beyond question.

The key way to look at RPA is to make constant comparisons to human administrators. Can they see and interact with the information? Can you give them rules (training) on what to do under which circumstances? Do they know who to ask and what to do if they get stuck? Can you put them in teams to help each other? How will you divide the job up to make it most efficient? The parallels are uncanny but thinking in these terms will help you determine whether the process you are considering is a good fit for RPA.



It is important to remember that in contemplating RPA projects, large processes need breaking down into smaller and smaller modules. In doing this we will find where the boundaries exist between the 80% of the work we can pass to the Bots and the 20% that must remain in the hands of humans. It is a good fit for an iterative, agile, approach as early efficiency benefits can be extracted by automating parts of a process, whilst building to completion.

In many cases, the temptation is to ignore systemic inefficiencies until they become unbearable. Then we embark on a major project to scrap existing systems and replace them; and we all love a shiny new system! Aside from the cost and delay this imposes, these new systems often introduce a whole new raft of process compromises. The introduction of RPA can help you get more from your existing infrastructure immediately, smoothing those inefficiencies and even help you migrate to a new system, when the time is right for you.

Look around your organisation. We are confident that it won't take long before you find people re-keying data, doing something so repetitive that it is mind-numbing, or that you come across an idea for an innovation that is stifled simply because the systems won't 'talk' to each other. When that happens, get in touch and we'll introduce you to organisations just like yours, which have been through the journey and have come out the other side. Then we can discuss what that experience can mean for you.

Further reading in this eBook series from NDL:



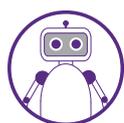
**API's: A Buyer's Guide**



**The Little Book of RPA**



**RPA Implementation Guide**



*This eBook is for informational purposes only. The information contained in this document represents the current view of NDL Software Limited on the issues discussed as of the date of publication. Because of changing market conditions, NDL cannot guarantee the accuracy of any information presented after the date of publication. Nothing in this document may be taken as a commitment on behalf of NDL Software Limited. NDL MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of NDL Software Limited. © 2018 NDL Software Limited. All rights reserved. NDL, the NDL and NDL logos are trademarks of NDL Software Limited in the United Kingdom and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.*

