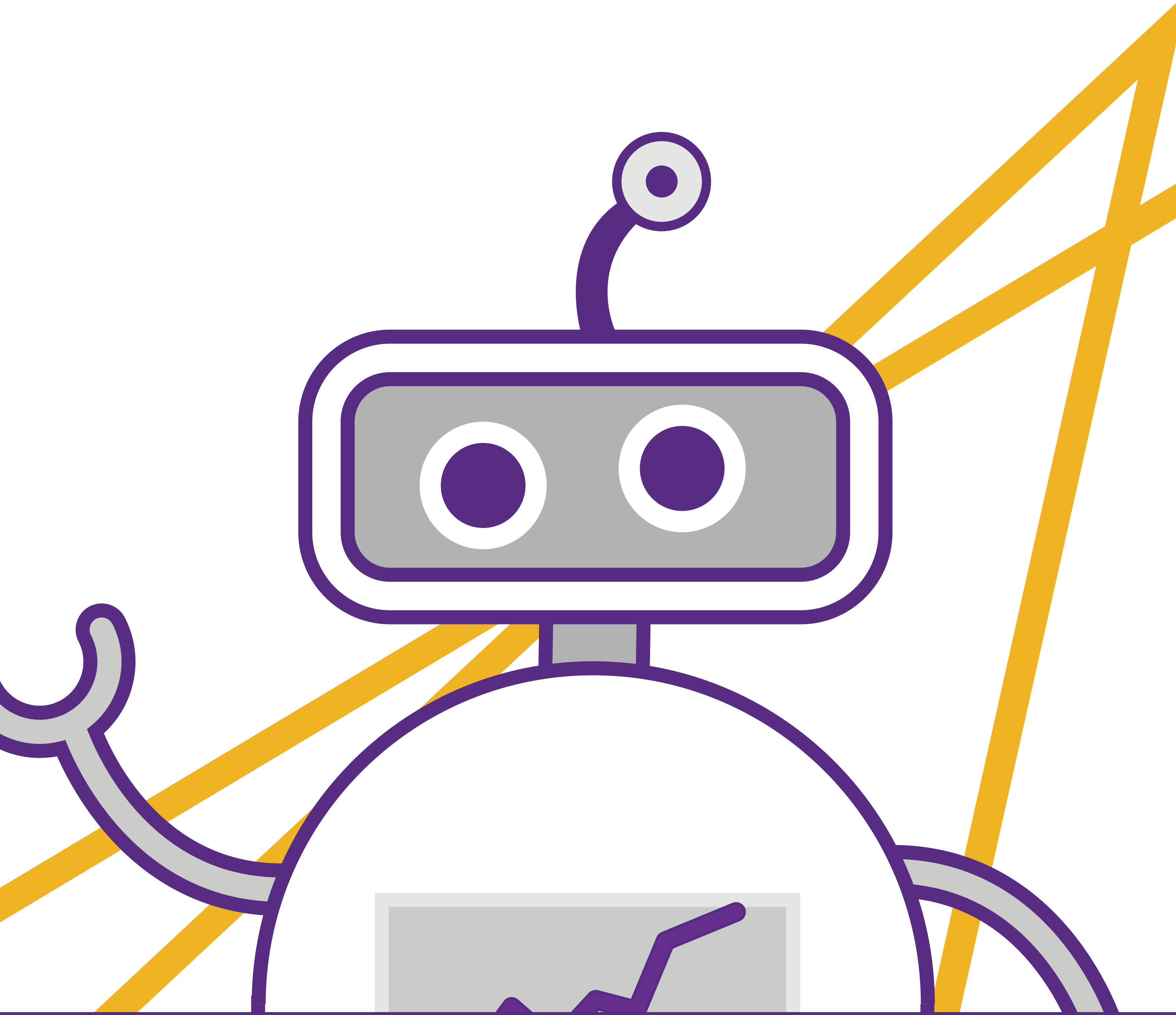
RPA Implementation Guide

Delivering A Successful RPA Project



NDL

Introduction

Robotic Process Automation (RPA) is software that allows users to develop and manage automations to artificially 'drive' any system, in exactly the same way that real people can. It does this at increased speed and with greater accuracy, often leading to a quantum leap in processing efficiency. In simple terms, it is like having an army of robots to do the work for you.

RPA holds the key to freeing up significant amounts of staff time, improving the quality, accuracy and availability of data records, whilst allowing organisations the time and freedom to redesign business processes to meet their changing needs. All this, but without being beholden to a siloed business mentality. Although RPA represents a great opportunity, organisations can find themselves tripping over the many pitfalls on the road of practical delivery.

Fortunately, these can be easily avoided and may, with hindsight, seem obvious. This document is designed to highlight these potential pitfalls so that they can be considered when planning your RPA projects. It will also suggest some ways in which you might mitigate those risks, hopefully saving you from repeating the mistakes of others.

A Note On Definitions and Background

This document is designed to be RPA specific, not Project Management 101. All the good practices that go into creating and delivering a normal project such as specifications, timescales, milestones, critical path, responsibilities, ownership definitions etc still apply but do not form part of this document.

Another point worth noting is that although RPA is usually delivered in two different ways, sometimes referred to as Attended and Unattended modes, we are mainly focussing on the latter. Attended mode is where desktop software (a Bot) is placed on a user's machine to deliver enhanced functionality or set to respond to a user circumstance, on the screen infront of the user, right there and then. Whereas many of the same points still apply and ought to be considered, we are mainly concerned with the 'lights out', Unattended, RPA element. This is where a 'Trusted Bot' works independently in a server based environment. Where a point specifically applies in either case, we will highlight this.

Finally, for context and especially for those new to the concepts of RPA, we will draw parallels between the Bot and the human worker. This is often a useful lens through which to view its implementation. Issues and mitigations can be identified by simply posing questions about what is reasonable behaviour for a human and what would we like a person to do in the same circumstances?

To RPA or Not to RPA - What sort of question is that?

Rather an important one as it turns out. Despite lots of claims about AI and Machine Learning, we are a very long way from Bots reliably being able to pattern-find and intuit the way a human does. Equally, on the plus side, Bots don't get bored, introduce errors, gossip or take breaks. Bots don't mind repetitive work. It is very much a case of horses (or Bots) for courses.

It is essential when setting out to make sure that all parties involved have clear expectations and understand what is and isn't reasonably possible. When considering if a project is suitable for RPA, we recommend that you think about the following questions:

Is the Process Consistent?

Do the systems or screens involved behave the same way all the time? For example, enter the account number into field 'A', hit 'Submit', the balance appears in field 'B', click the 'New' button – and Repeat. If the system you are showing the bots behaves in an inconsistent manner your training curve is going to be longer, as it would be with humans. Complex processes can be made up of a series of smaller processes, but the behaviours of the underlying components need to be stable.

Are the Business Rules Well Defined?

Where we are making decisions based on logic and algorithms, we need to be extremely clear about those rules. For instance, we may introduce a rule that says if the Pathology results registered in a field are 'Clear' then send a notification, if not, refer to a different clinician for diagnosis. A nice, clear and consistent rule: So, rules-based rather than judgement-based.

Whatever the press and leading-edge IT companies may wish us to be excited about, we are a long way from AI or Machine Learning being able to interpret many pages of hand written notes to determine a diagnosis.

Bad Automation Targets

When human intuition is required

Inconsistent Process

Low Volumes

Infrequent Process

If you cannot break the steps down into simple steps and rules, then the project is probably not suitable for RPA. Don't forget, however, about the 80/20 rule. You may well be able to get the Bot to manage 80% of the transactions automatically, as long as you have a referral mechanism for the other 20% - still saving the organisation a vast amount of time.

Volumes

It may seem obvious but RPA is best suited for high volume transactions or, more precisely, those that consume a significant amount of human time. If something happens or needs doing just once or twice a day, then it is unlikely that there will be a sufficient ROI. We raise this as we have seen projects of this nature being attempted. There are always exceptions such as out of hours or time sensitive cases, however, having a clear view during the planning stage of the time that will be saved will help with the demonstration of ROI and contextualise the investment in the project.

Structured vs Unstructured Inputs?

Ideally data needs to be in well-defined digital forms, such as fields from a database or web service, rather than trying to extract the data from free-flowing text.

Human Error

Processes that involve high error rates when data is entered manually, such as those that involve lots of rekeying, can be ideal targets as the accuracy of RPA can avoid these.

Compliance

Where compliance is an absolute 'must' and non-compliance would lead to fines or damage, then RPA based approaches can ensure that all the necessary elements are completed and none are missed.

Good Automation Targets

High Volumes

Logical

Consistent

Multiple Source /
desination systems

We are looking for those high volume transactions that are consistent with well defined business rules. A key point here, however, is that this does not mean that complex processes cannot be tackled. We would just recommend that they are built up through the orchestration of a series of simpler transactions.

A further consideration in the justification of RPA, and one that adds an extra dimension to the business case, is a process where multiple systems are involved. When done manually, these processes tend to involve a considerable amount of staff time and training, and switching between systems tends to introduce an error overhead. This scenario would be the perfect example of when RPA automation is ideal.

Perspective

A useful way to look at RPA is to make constant comparisons to human operators. 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 do they know when to start and how to finish? How will you divide the job up to make it most efficient? The parallels are uncanny, but constantly thinking in these terms will help you determine whether the process you are considering is a good fit for RPA.

Getting Off To A Good Start

It is always tempting to 'dive in', particularly where tools ostensibly seem to make it easy and where quick results can been seen. However, taking the time to create a careful thought-out specification will pay dividends throughout and beyond the life of the initial project. We have often seen initial self-training or experimental processes morph into live transactions that perform less than optimally and ultimately need redesign. Therefore prototyping and the proving of concepts have their place, but they should not be confused with production.

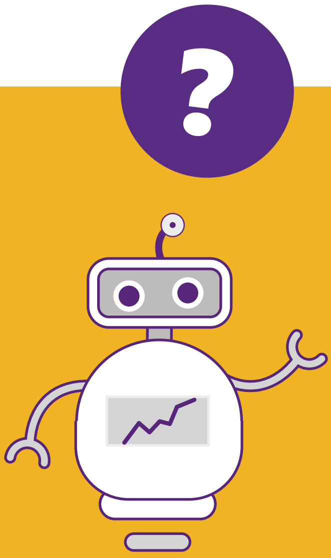
As mentioned above, we are not going through the whole process of managing a project and the separation of Requirements/User Stories vs Design documentation etc here. These should of course be done to whatever level is deemed appropriate, and will no doubt be invaluable in managing, testing and documenting the process and project. At a more practical level however, in an RPA process, one essential component ought to be a 'storyboard' and workflow process map of the manual steps required.

The Storyboard should follow the process and be compiled with the assistance of an Expert User who has detailed knowledge of the usage of the systems on a day to day basis. For the avoidance of doubt, we are referring to an end user, not an IT admin, if at all possible. It should include images of the screens involved, highlighting controls (buttons, fields, images etc) of interest and explain their context and any navigations. Attention should be particularly paid to interruptive behaviours, warning boxes, pop-ups, error messages that may appear. As part of this definition, it is also worth understanding the end user view of the stability and reliability of the systems involved, noting any regular or common down-time or periods when performance is poor.

At this point, it is also very important to take note of the amount of human time taken to perform this transaction in a realistic setting so that a project baseline can be established. In addition, any compliance risks and benefits should also be recorded. The number of transactions and any expected growth or shrinkage should also be noted. – This is important in understanding the overall financial viability of the project.

We would also recommend that the wider views of the Expert Users are sought; such as on the regularity of updates or upgrades to the back-office systems and their effects on the User experience. Also in a Public Sector or Financial Services arena, are there any pending legislative changes that may impact the process? Does the user feel the process could actually be improved or additional value added? This of course needs to be done in the context of the transaction making up just one strand of the overall project. It may simply be a case of getting an explanation of the 'as is' process rather than the desired outcome. The fact remains however, the better understood the process, systems and environment involved, the shorter the development and better the outcome.

To some much of this is obvious. However frustratingly, we have seen more than one instance where the end user had a detailed specification yet the implementation team and their contractor had never seen it. Furthermore, certainly in one instance it was clear from the specification that the requirement could never be achieved and should therefore never have been attempted.



Attended and Unattended RPA. Whats the difference?

There is a key differentiation here between projects involving 'Attended' and 'Unattended RPA. In the Unattended model, where the system is being run solely by a 'Trusted Bot' in a lights-out 'virtual' environment, you can control that environment to the nth degree, making sure that there are no interrupts, unplanned updates etc. However, in an Attended environment, external elements often have the capacity to interfere with the Bot, whether systemically or user created. The propensity for this will depend very much on the level of corporate control exercised over the desktop and application environment. These are not insurmountable, just another point to bear in mind when either approach is considered.

The Beginning of the End?

Having examined the target process, story boarded it and agreed that it is both possible and justified, there are two further elements that need to be closely defined. These are; 'where does the process start and finish?' or 'Why and how does the process start and how does it finish?' The best way of articulating this is by examining some practical scenarios, starting with how does the Bot process *actually* trigger?

Some examples in an Unattended environment are; a new record being written in a database, a file being saved into a folder, the arrival of an email or a post to a web service. The system can monitor these and trigger the Bot to respond in your chosen manner. The result of that activity may simply be restricted to entering data or manipulating one or more target systems. It might also involve a web service response, database entry etc. Being clear on this, the necessary data elements, formats, addresses and permissions in your specification are crucial. We have considerable experience of project complexity lying in this area rather than being anything to do with the actual robotics process, so do not gloss over this area.

In an Attended Automation the same principles apply, however, processes are more usually started by either a user button that has been added to an app or desktop, or the RPA starting as the result of a trigger, trap or monitored conditional change on the user desktop / application. Aside from any data elements, inputs and outputs, it is also important to consider how you wish the user to behave whilst the Bot is in action. Do you wish for example to suspend user activity on the target system, like autopilot in a car? Is the bot invisible to the user, simply recording and posting information for use elsewhere? E.g. Trapping a change or circumstances and posting the new details to a web service. Do you want to inform them or not? All points worth considering when defining the process and specifications.

Complexity, the Enemy of Delivery

Almost every process should be broken down into logical units or sub processes wherever possible. This has several benefits. The obvious one is that what may have seemed impossible before now becomes quite achievable. Furthermore, the organisation can often get benefits earlier rather than waiting until the end of a long waterfall style project. There are however several other benefits that come to light too, when this approach is taken.

The less complexity there is in a process, the less scope there is for error and if something does not behave the way you want it to, it is just easier to spot and isolate the cause. Beyond this, many long processes use the same sub-processes such as, logons, search, look-ups or change of circumstances, to quote just a few examples. These can then form a library of trusted processes that can be used in repeated projects, saving considerable process development time.

Also, paralleling the real world, different tasks are often allocated to different team members. The best example of this of course being the difference between the old-style one off artisan coach builders of the past and the innovation of Mr Ford and his production line. By breaking down the larger process into smaller units, these can often be allocated to different Bots who will work as a team. This brings flexibility, scalability and resilience to the operation as the Bots can be distributed into logical teams rather than having to be on the same infrastructure.

RPA, A Game of 2 Halves

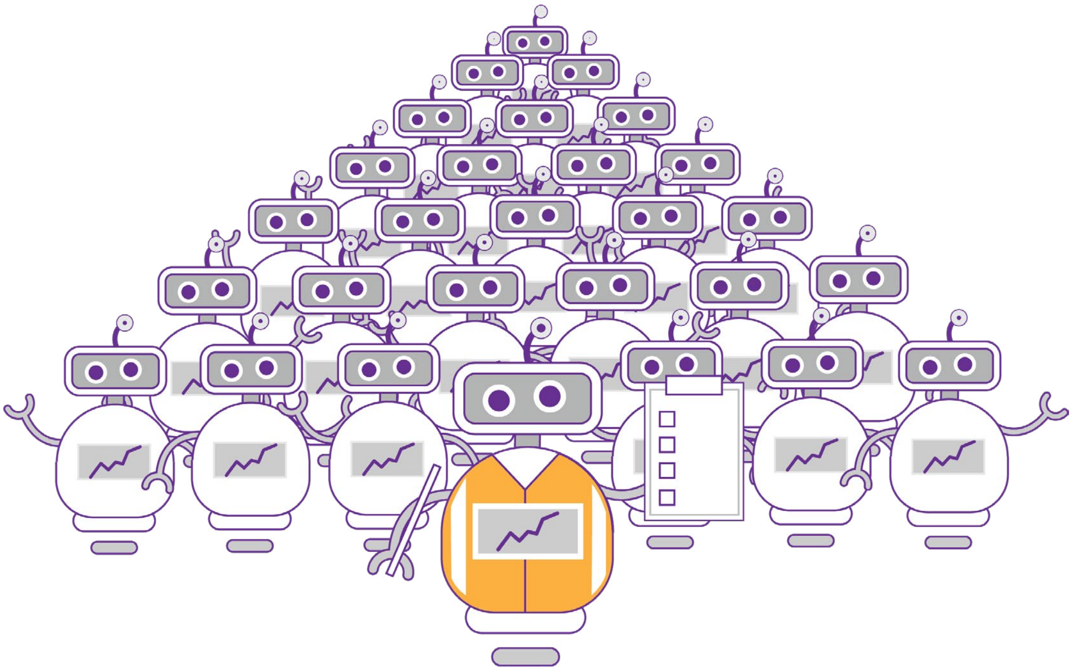
Earlier we discussed the Expert User and storyboarding. There is however a deeper consideration. The art of RPA is made up of all the process and technical understanding of the selected tools but you should not underestimate the need for a deep understanding of the back-office systems involved. We are not talking about what they do or how they do it, more the construction and behavioural foibles of those systems.

There are some commonalities between different web based apps, or Windows desktop ones and although we can learn some generic rules about using them, the internals and behaviours of a big back office system can vary widely. This is a consequence of those systems evolving over time, different development teams being involved in their creation, maintenance and the sometimes-patchy introduction of newer technologies or UX standards. Sometimes areas of a system that look the same may well respond differently to an RPA solution. That is not to say that the RPA will not work, rather that the most efficient approach in one area may not be the best for another.

Whereas there will always be new systems upon which to apply RPA, it is important, wherever possible, to select an RPA vendor with knowledge of your types of back-office systems. A vendor who can offer *contextual* rather than general advice about the target applications and environments concerned.

Orchestration and BI

If we are breaking individual processes down between a team of Bots or have multiple processes running, especially where they 'feed' each other, it is important to consider how you are going to co-ordinate the activities of the various bots, interfaces and processes. To this end, RPA projects often need some form of orchestration.



There are many orchestration engines in the market place, however, these tend to be highly complex, big ticket and intense computing systems that are the preserve of the expensive specialists. Ensuring your chosen solution can interwork with these is important and, in fairness, most RPA products do. Some however offer their own closely-coupled, simpler and lighter weight offerings, which can provide tremendous benefits beyond just co-ordinating the core RPA process.

Firstly, they can draw attention to systemic issues, for example; stalled back office systems, poor and rejected data. Secondly, orchestration queue management can play an important part in the concept of 'failing safe', knowing what has and what has not been successfully processed. Finally, and possibly most interestingly, as they tend to sit in the middle and have access to all the inputs and outputs of the business process, they present the opportunity to create rich data repositories. Business Intelligence and analytics tools can be used against these repositories to provide insights into the day to day performance of an organisation.

In an Unattended RPA project, consideration of closely coupled orchestration should be a must.

Be A Good User

Aligned to the expertise in the back-office system, is the concept of being a well-behaved user. Bluntly, most RPA offerings can completely over-drive most systems. They are simply so much faster than any of us and usually faster than the response time of the target system; especially when a team of Bots is hitting it hard.

When developing your process make sure you consider this point. Try to understand the limits of the target system and think about tidal flow and scale - building controls into your process as a result. A good RPA offering should allow you to interact and respond to the events in the system, rather than using brute force to feed it or having to build in timers to make sure that the application is not overrun. Ideally it should be able to match the flow of transactions with target system performance. It's the robotic equivalent of being a well behaved and balanced user rather than one who gets frustrated and mashes at the keyboard.

One way to approach this, apart from having a system that is able to respond to system performance and availability, is to consider scheduling your use of Bots to take advantage of known slack time such as overnight. Obviously, the ability to do this will depend on the transaction latency, but it is worth considering in process capacity planning.

Testing Time

It ought to go without saying that before going live there should be a thorough test process, looking at different failure conditions and how your processes respond. We guarantee that even when live, something will go wrong. Even if it's as simple as the cleaner unplugging the back-office server to plug in the vacuum. (We jest not!). If we accept this, there are several things that must be considered. Firstly, can we reduce the likelihood? Secondly, how do we 'fail safe'? and thirdly, how are we going to respond in the event to an issue once it is past? A further allied consideration is: how do the Bots notify us if there has been an issue? and what level of audit log do you wish to keep for diagnosing issues or validate transaction processing?

The simplest mitigation here is testing. Testing against the specification, stress testing, deliberate failure testing – it's an art in itself. In our experience it is overlooked and underdone in the rush to 'get something live'. In RPA projects, only a third of the time should be spent on creating the process, the rest should be applied, in roughly equal measure, to specification and testing. Indeed, the better the specification, the clearer the testing needs become. Inclusion of a section on performance and testing should form part of that document.

Audit and Management

Once the creation and deployment stages of your RPA environment are complete and you move to 'Business as Usual' (BAU), management of the environment and processes becomes more of a concern. Just as you would expect the manager or team leader to be able to monitor performance and sample KPIs, the same is true in the RPA world.

Operational visibility and control are just as important as creating the specification and testing, after all, these are likely to be key transactions with important organisational data. We would therefore suggest that having a comprehensive management suite offering the ability to monitor performance, introduce and retire individual Bots or Bot-Teams, control the processing of transaction queues and any parameters surround them, is key. Even allowing them to be switched between production servers and environments.

Along with this goes audit. Security (below) is often cited as a concern, therefore, being able to retain comprehensive transactional audit data can be key. This should be exportable in a useable form, under the correct control of course, and the granularity of the collection of that data should also be fully controllable.

In a modern DevOps environment, we do not sit with our management systems open, as this is in itself a security concern. To this end, the ability of your BAU management system to be able to alert the DevOps team in a number of different manners is also useful.

Finally, if considering an Unattended, lights off, RPA solution that will scale, the ability to load share, failover and move transactions in an environment that may span on premise, or cloud environments, should also be considered. We would also urge the consideration of a parallel 'test' system to help with a staged change control process. This will all need to be relatively simple to achieve and should be visible to the DevOps team.

Security and Trusted Robots

Security is an issue that cannot be ignored. RPA is treated with suspicion by many, usually due to a lack of knowledge. Security concerns are often used as the last bastions of the unwilling, to frighten the ill-informed. However, in our experience, Software Robotics is a far more secure route than comparable human teams, in almost every aspect.

Firstly, as it is usually task-based, the environment can be locked down far more tightly than any PC being used by a real person. The robot will only do exactly what it has been instructed to do, nothing more, nothing less. There is no screen for someone to watch. Instructions and data passed to and from the robot are encrypted, unlike the instructions given to a human. They don't make typos or gossip and everything that is done can be rigorously audited at a level unachievable by human management.

Security is a concern and needs to be considered as part of the specification, forming part of the testing (and re testing) regime. In the final analysis, an RPA approach is highly unlikely to be weaker than any comparable alternative method of processing your business transactions. We have certainly never seen an issue, but secutiry issues need to be acknowledged by the stakeholders and then built in by design to the tools used and the processes created. It also needs viewing in the context of alternative approaches. It is almost impossible to produce a 100% perfectly secure system of any nature. The real question is: is it fit for purpose? and is better or worse than the alternatives?

Change Is Inevitable

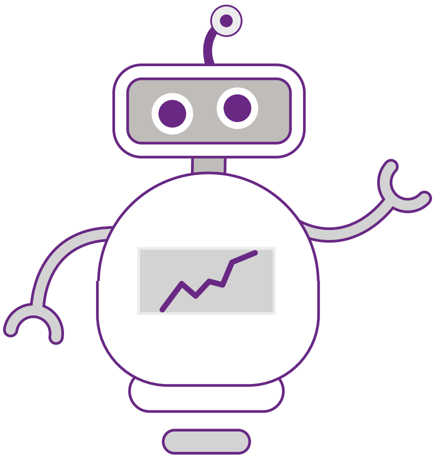
Business processes always change, especially in today's fast-moving environment. Being aware of, and planning for change is essential in maintaining your BAU state. It is important, therefore, that there is a well understood change management process and that changes are not just implemented on an ad hoc basis, without planning and proper testing.

Aside from process changes and the inevitable underlying operating system updates, sometimes back-office systems change too. In our experience, it is rare for a system to change dramatically. This happens maybe once a decade or so, when a vendor adopts a new technology; such as moving from a Windows to web based delivery. More likely are incremental changes to the system functionality. The good news here is that this usually has very little effect on existing processes, unless major new functionality is being introduced or change is driven for some new legislative reason.

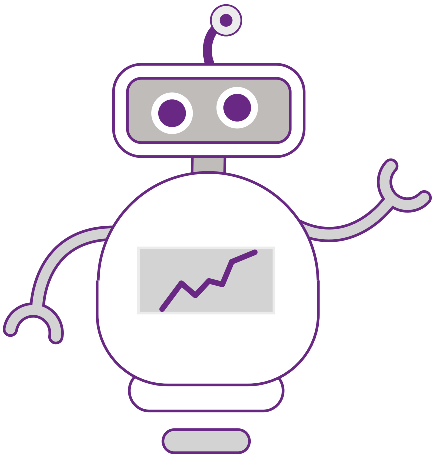
Users hate change and unfamiliarity so vendors generally avoid this, especially as organisations do not want to have the expense of re-training their staff. What tends to happen in reality is that minor functionality changes are just added, for instance: the inclusion of an additional field. These additions tend to be at the end of a screen or in a new section, rather than interfering with the flow of the existing layout. Your 'Expert User' should certainly be able to give you guidance on how regularly, if ever, the back-office application gets updated and if any major new version releases are being contemplated by vendors. These are usually the subject of much noise years in advance.

Try to predict and plan for changes so there are no surprises. Work with vendors to try to anticipate back-office changes and any effect they might have on RPA processes. As the title suggests, change is inevitable, but it is also easily manageable as long as a suitable process is in place and it is adhered to. It certainly should not be ignored.

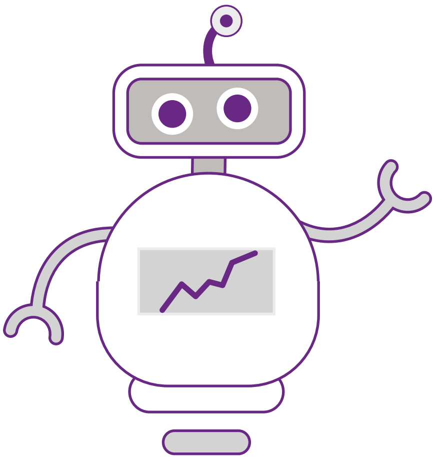
Building an RPA Transformation Team



We believe that an organisation and the people in it ought to be the best qualified to apply RPA techniques to its business processes. Despite the size of this document, the technology and creation of an RPA process is not all that difficult. It is all the ancillary business issues surrounding it such as the process definition, that takes the time. Due to this, there is no one better placed to understand those processes than the business itself.



To that end, we would recommend the creation of a team experienced in implementing RPA techniques. As soon as the first couple of processes are delivered and the results are demonstrated, it is highly likely that more and more areas of the business will show an interest in its application. Selecting the right tools and having your own team to implement it is in our view, a must. The cost and time lag involved in large-scale outside consultancies can make the bar so high that nothing can ever be justified, except in the case of extremely large projects, where the cost of change can become prohibitive.



It may well be that you turn to the vendor or a partner to assist you with your first project or two, but those should be a means to a self-sufficient end. In that regard, the availability of training, good documentation, support and possibly a like-minded user community, all become key differentiators when choosing who to work with.

Record and Prove Your ROI

So, you have a project implemented, it's all business as usual and you are now no longer re-keying lots of data. Happy days. Very quickly, however, this becomes the norm and the base line for business performance.

Earlier in this document we discussed establishing the amount of time people would take to perform a task manually. It is important to make sure you can report on transaction volumes and the true value of the project to the end user department.

What often happens is: the IT team carry the cost of production, licences etc and the business department receives the end benefit. If, however, there is going to be management buy-in for ongoing support or for future projects, then this transactional ROI information needs to be visible and reportable. It must also be on a realistic basis. Merely translating 10 minutes a transaction into 6 transactions an hour and saying that one FTE can therefore do 45 per day, is unrealistic. There are enough studies to show that humans only run at about 60% efficiency in a day. Then you need to factor in overheads, holidays, sickness, etc. So, for every three FTE's - a fourth is needed for cover and then a manager, let alone office overheads.

These should all have been noted in the commercial business case, but it is important to be able to evidence this to the business, based on real transaction data. As such the ability to report on this simply and regularly should also form part of the RPA tool set choice.

Summary

There are many points listed in this document and it may seem daunting. We hope, however, you will agree that in the majority of cases, they are not rocket science. Indeed, there may be a temptation to short cut or ignore some of these points. We would urge against this. Take the time to do it right rather than having to do it again. If done well, the benefits of your successful project will, by far, outweigh the time spent on it. So even if you feel that an element does not apply in your case, take the time to document why this is... You may be surprised.

We have been privileged to work with a wide customer base for over a decade delivering RPA tools and projects. This has seen:

- The liberation of data from locked up systems allowing clinicians to use predictive analysis to prioritise patient pathways
- The collection and aggregation of mapping data to allow those working on construction projects to have a safer working environment.
- The creation of new services and workflows within Local Authorities and Financial Institutions, to process billions of data elements which would previously have to have been done by hand.

The benefits of well executed RPA projects for an organisation are truly amazing and often not appreciated until the journey starts. Whatever toolset you choose to use, we hope you have the very best start, and hope that the points raised in this document help you with that.

Start Small

Clear storyboard/specification

Measure

Have a change process

Be clear on ROI

Break it down to small steps

Test, Test, Test,

Compare to human approach

Make a start

Orchestrate

If you would like to explore this topic further, why not download the other documents in the series

An Introduction to RPA

The Little Book of RPA

The API Buyers Guide

Or, you could get in touch as we'd be happy to discuss your needs and share our offerings and experience with you in person:

Email: info@ndl.co.uk

Tel: 01937 543500