



NOVEMBER 2019

**ALL THE WAY WITH**

# **RPA**

**[BETTER TECHNOLOGY FOR YOUR BUSINESS]**

**AUTHOR:**

Philip Radburn

# BETTER TECHNOLOGY FOR YOUR BUSINESS

**Robotic Process Automation (RPA)** is exciting technology that is gaining widespread use, going the way of other software such as ERP and CRM.

As with those applications, some RPA projects have been disasters and there will be further failures as adoption continues. Failure can be avoided however if you consider four (4) key questions:

1. **Why build/buy anything?**
2. **What is the right thing to build/buy?**
3. **How do I build it right?**
4. **How do I use it right?**

To answer the first question, you must consider the full gamut of options – from better use of existing technology, through total disruption of the existing metaprocess.

To build/buy the “right thing” you must clearly understand:

- **The diversity of stakeholder groups (from users to auditors)**
- **Their diversity of functional requirements**
- **The range of capabilities they require.**

Any piece of software requires some logic development (“coding”) and it is during this “build/buy it” stage that things often go wrong – in terms of cost, schedule and quality. Critical to you “building/buying it right” are:

- **Shifting from spending on finding and fixing problems to investing in preventing them**
- **Use of a development/implement approach that is aligned with overall project goals**
- **Use of a broad set of methodologies to support the approach and avoid defects and their impact on budget, schedule and quality.**

Finally, “use it right” means more than operating the software. Your whole organisation – the user team(s) in particular – will be affected by the new solution and you must carefully manage this change to ensure you seize any emergent opportunities.

# 4 KEY QUESTIONS

**Why Build/Buy Anything?**

**What's the Right Thing to Build/Buy?**

**How do I Build/Buy it Right?**

**How do I Use it Right?**



# WHY BUILD/BUY ANYTHING?

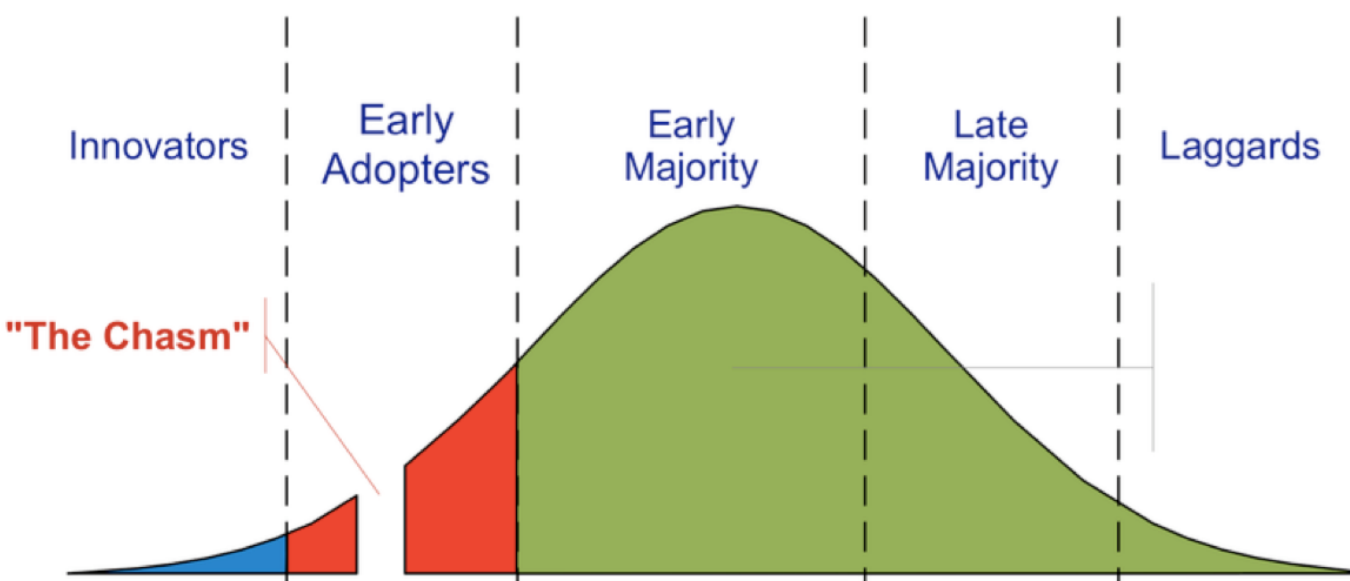
## WHY DO ORGANISATIONS CHANGE TECHNOLOGIES?

**Coercive factors** - pressure from governments and corporate functions and the expectations of society

**Normative factors** - stem largely from the professions (for example, accountants adopting the software recommended by their professional association)

**Mimetic factors** - adopting solutions already in use with similar organisations.

Organisations that want to leverage the experiences of others do so when the technology has “**crossed the chasm**”.



# CURRENT STATE OF RPA?

Gartner reports that RPA is the fastest growing software segment that it tracks. RPA's success is beyond question according to Forrester and users are now wanting to move to broader use cases.

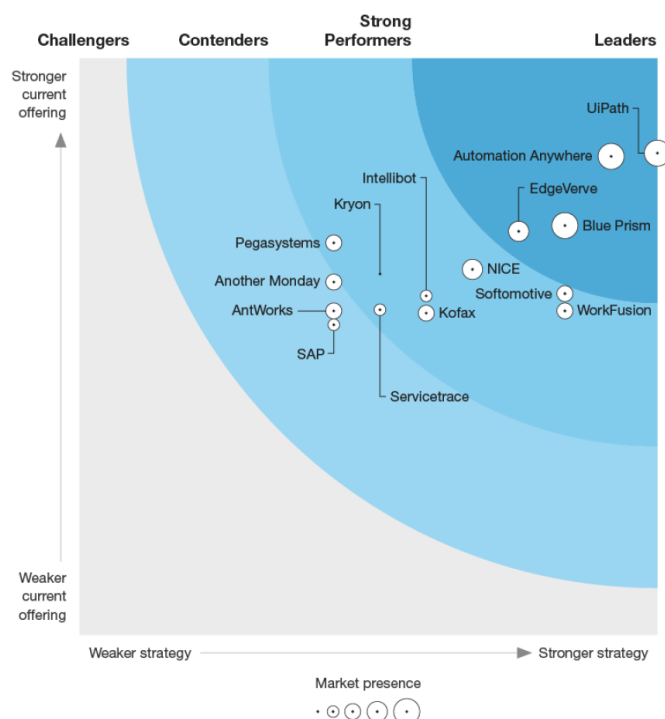
Big consultancies have set up RPA service lines and, within companies that have a year or two of experience with RPA, champions are looking for further viable use cases.

All the above indicates that RPA is "in The Chasm". Will it cross it? Will your organisation become one of those that help it succeed in general adoption? To do so, you'll need to answer the question "why should I change"?



## TIP

*Gartner and Forrester research is sometimes available free of charge from vendor websites. McKinsey also writes about RPA and its articles are available on their website.*





# WHY RPA?

RPA can be used for processing a transaction, manipulating data, triggering responses and communicating with other digital systems.

That is, there are wider uses for RPA outside business process automation, for example, monitoring computer networks for cyber-attack and initiating action.

The critical aspects of robots include **quality** and **performance** attributes as well as **function**.



## TIP

*Gartner advises to watch out for:*

- *Applying a robotic Band-Aid to a particular pain point, rather than take an end-to-end view of the outcome needed and measure that the delivered output is better.*
- *Shifting of manual process debt to technical debt*
- *Undervaluing human process knowledge that, and once programmed into the robot, disappears.*



## WHY NOT RPA?

Mostly, RPA is being used to address "pain points" or bottlenecks in a key business process. The symptoms may be poor customer service, low revenue, high cost or high risk (such as cyber-attack and fraud) and RPA may be very suited to their mitigation or elimination.

However, other viable alternatives may also be available.

### ALTERNATIVE USE OF FUNDS

Adopting RPA may involve substantial investment that might be better spent elsewhere (including outside IT).

### BETTER USE OF CURRENT IT

Technologies (ERP, CRM, etc.) are often heavily customized. The standard applications also tend over time to gain additional features, offering greater capability than that currently in use. Before embracing any new technology, such as RPA, you should examine your use of current technology to ensure their implementation and maintenance have not created inefficiencies and delays.

### ADDRESS THE CAUSE OF PAIN

Major processes, such as procure-to-pay and hire-to-retire, will inevitably involve multiple functions. In this environment delays, waste and inefficiencies are rife and often one activity may be a bottleneck.

### DISRUPT THE PROCESS

It may be that the bottleneck you're considering for RPA is one of many significant sources of waste and ineffectiveness in an end-to-end process. It may be better to consider eliminating them all in one fell swoop

# WHAT'S THE RIGHT THING?

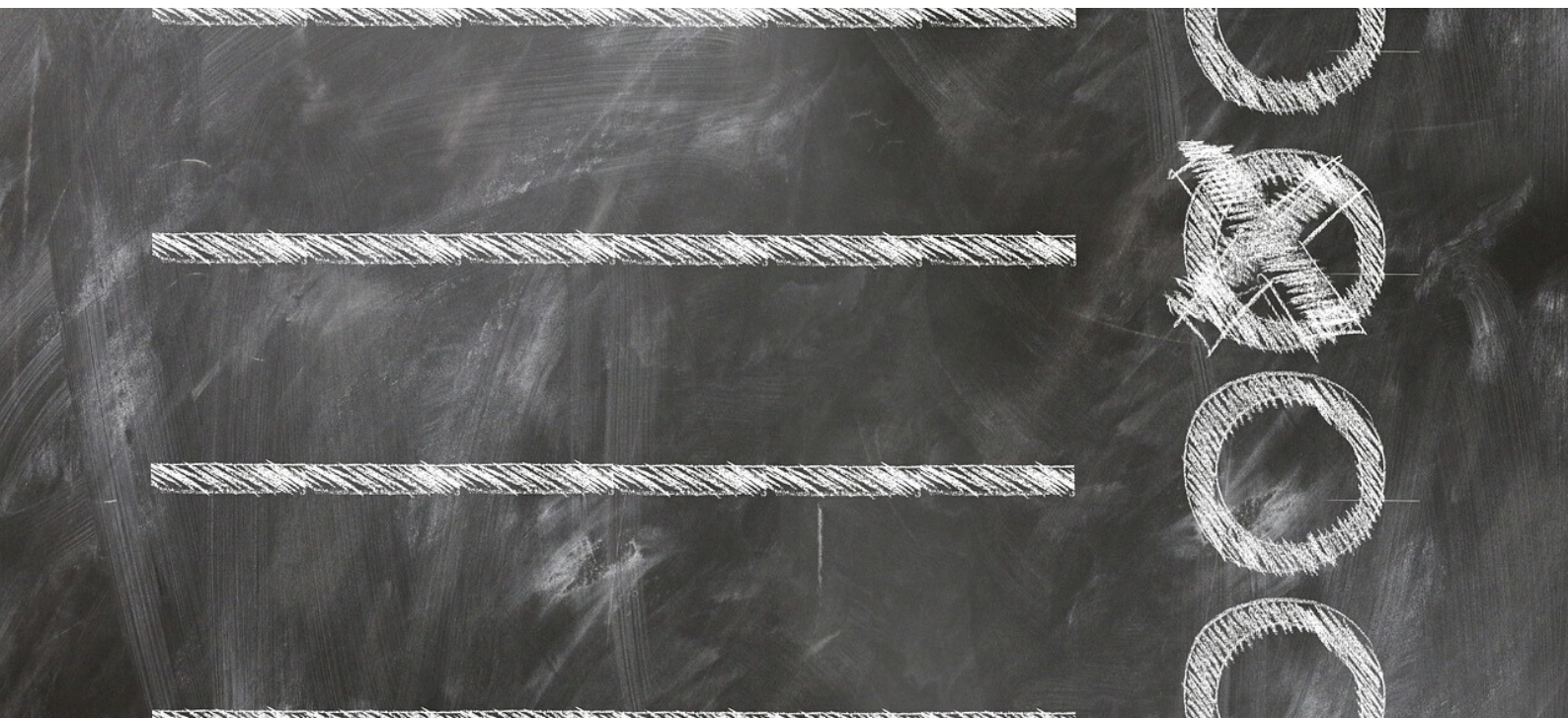
Whatever approach you use to build/buy your RPA solution, it will undoubtedly have some form of “requirements specification” phase or activity during which the “Right Thing” is defined.

There are 3 dimensions to this definition and to Build/Buy the Right Thing you must cover all three:

**Stakeholders** - Whose needs are to be met?

**Function** - What are their needs?

**Attributes** - How well must their needs be met?





# STAKEHOLDERS

A common problem in software development is a focus primarily on the “end user” of the software and their needs, but any piece of software has many stakeholders.

**User** - the person whose activities are being augmented or supported by the software (robot in the case of RPA)

**Operator** - schedules the software (robot), ensures it is functioning and addresses any exceptions or alerts

**Auditor** - confirms the design and operation of the software (robot) for compliance with policy, regulations, etc

**Installer** - takes the software (robot) from its developers and introduces it into operations

**Maintainer** - enhances and fixes the software (robot) and related files

**Funder** - pays for the design, development, operation and maintenance of the software (robot)

**Planner** - ensures the software (robot) continues to meet the needs of the other stakeholders once it enters operations.



# FUNCTIONAL REQUIREMENTS

A functional requirement is an essential activity that a solution must perform (for a stakeholder). Functional requirements are often divided into categories to support a particular goal.

For example, in Agile projects, they are categorised in terms of delivery priority - **Must** have, **Should** have, **Could** have, **Won't** have (**MoSCoW**).

Other categories include performance impact, likelihood of change, and technical risk.



## TIP

*Although this may seem obvious, it is important to get the appropriate level of detail in your functional requirements for your solution – not too detailed that it takes excessive time to document; not too high-level, complex or vague as to risk omission or error during development.*



# ATTRIBUTES

Attributes are related to the functions of the solution and the stakeholders that require that function. An exhaustive list of system attributes is available from a number of online sources but some key capabilities are:

**Usability** - the ease with which a user can learn to use the software and interact with it in operation,

**Operability** - the ease with which the software can be put into operation and managed

**Security** - the ability of the software to withstand unauthorised use, information loss and system damage

**Reliability** - the capability of the software to maintain its required level of performance

**Availability** - the degree to which the software remains usable and operable

**Maintainability** - the effort and time required to make changes to the software.

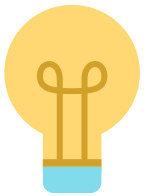


# TRACEABILITY

Once you have a complete and correct list of stakeholders, functional requirements and attributes you need to bring them all together to get a clear picture of which stakeholder(s) have a need for what functions and to what level of quality and performance.

This is established using a traceability matrix. Such a matrix is a simple tool for detecting inconsistencies (incompleteness and incorrectness) in your captured requirements.

The matrix is the baseline for ensuring you “Build it Right”.



## TIP

*If your RPA program is small and simple, you could use a spreadsheet to manage traceability; if not there are many tools available and your software vendor may also have an offering that does the job.*



# BUILD/BUY IT RIGHT

RPA projects are effectively custom software engineering projects. Some products and tools, and some uses of them, require more logic development than others, but all require some software engineering activity.

This activity will usually (very risky if it doesn't) follow a defined approach based on a "philosophy" of software engineering such as Waterfall or Agile.

Key software engineering concepts that must be considered in an RPA project:

- **Software Cost of Quality**
- **Software Engineering Approaches**
- **Software Engineering Methodologies**



## **TIP**

*RPA is often taken up by business units before their IT organisation. Although there are certainly advantages to this approach, leaders of these initiatives are advised to seek the advice and/or assistance of a qualified software engineer if extensive use of RPA is envisaged.*

# SOFTWARE COST OF QUALITY

Building any piece of software entails costs. People's time is a variable cost and often "blows out" during a project. The bulk of "people time" on an software project is spent on designing, coding and testing software logic.

Although minimal logic development ("coding") may be required in simple applications of RPA we are not yet at the point of end-to-end automated software engineering.

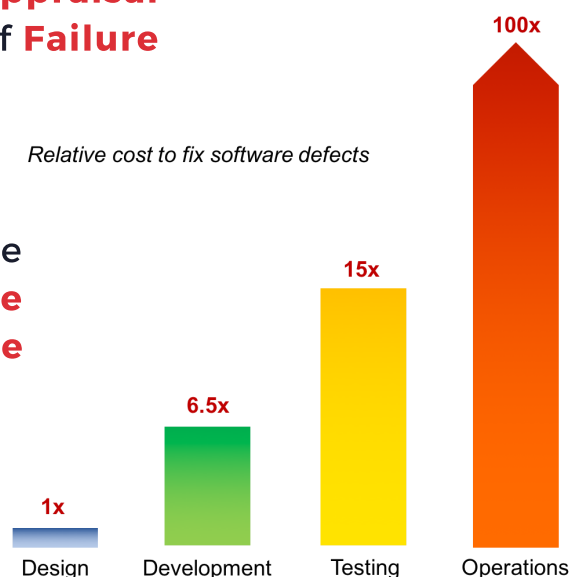
The core "building" activity is coding. Design work confirms "the right thing" to be coded and testing is meant to confirm that it has been coded "right". However, testing usually becomes an exercise in finding "defects".

As it takes developers the same time to write the erroneous code as it would take for them to write correct code, you can see why software costs nearly always exceed budget.

The Cost of Quality of software is the sum of costs related to prevention and detection of defects and the costs arising from defects. This is usually expressed as:

$$\text{Cost of Quality (COQ)} = \text{Cost of Prevention} + \text{Cost of Appraisal} + \text{Cost of Failure}$$

*Relative cost to fix software defects*



The fundamental axiom of the software cost of quality is that **the later a defect is discovered the more expensive it is to be fixed.**

# SOFTWARE COST OF QUALITY

## COST OF FAILURE

The cost of failure is the costliest of the three components of COQ and has two dimensions:

**Internal** – from defects discovered before the product is operational, and includes rework of logic

**External** – from defects discovered after the product is operational; incurred by the project team and stakeholders, and includes rework, warranty claims and delayed benefits.

## COST OF APPRAISAL

These costs arise from efforts to find defects introduced during development/implementation. The costs of appraisal include inspections, reviews, audits and testing.

Testing is a wasteful: if there are defects, you've created them and now you will have to correct and retest them; if no defects, then you may have wasted effort in test planning, design, conduct, review and reporting!

## COST OF PREVENTION

Investment in prevention gives the greatest return in terms of achieving the project cost, schedule and quality goals. The minimisation of the costs of (poor) quality through prevention is driven by two critical elements:

**Early prevention activity** - includes training team members and developing standards

**Broad prevention activity** - includes configuration, knowledge and change management.

# SOFTWARE ENGINEERING

## APPROACHES

**Waterfall** and **Agile** are the two most common approaches likely to be employed for an software project but there are many examples of the use of each where the project has been deemed a failure.

## METHODOLOGIES

Although organisations that conduct IT-related projects will have a project management methodology there are many more methodologies that are critical to any software project (including RPA). These include design and development, verification and validation and quality management.

The primary reasons for using methodologies are:

**Risk reduction** – they embody the “investment” approach to managing the cost of software quality (above)

**Attribute realisation** – capabilities such as maintainability, portability, and security are more readily achieved if your methodologies incorporate your organisations policies in respect of these.



### TIP

*If you have, or intend to, invest heavily in RPA, spend some effort deliberating on the various approaches and methodologies and how their tenets may be best implemented in your business and IT environment.*



# BUILD/BUY IT RIGHT

## BRINGING IT ALL TOGETHER

Until we have fully automated software engineering an RPA project will present serious risks of functional incompleteness and incorrectness, capability that is below the expected and budget and/or schedule overruns.

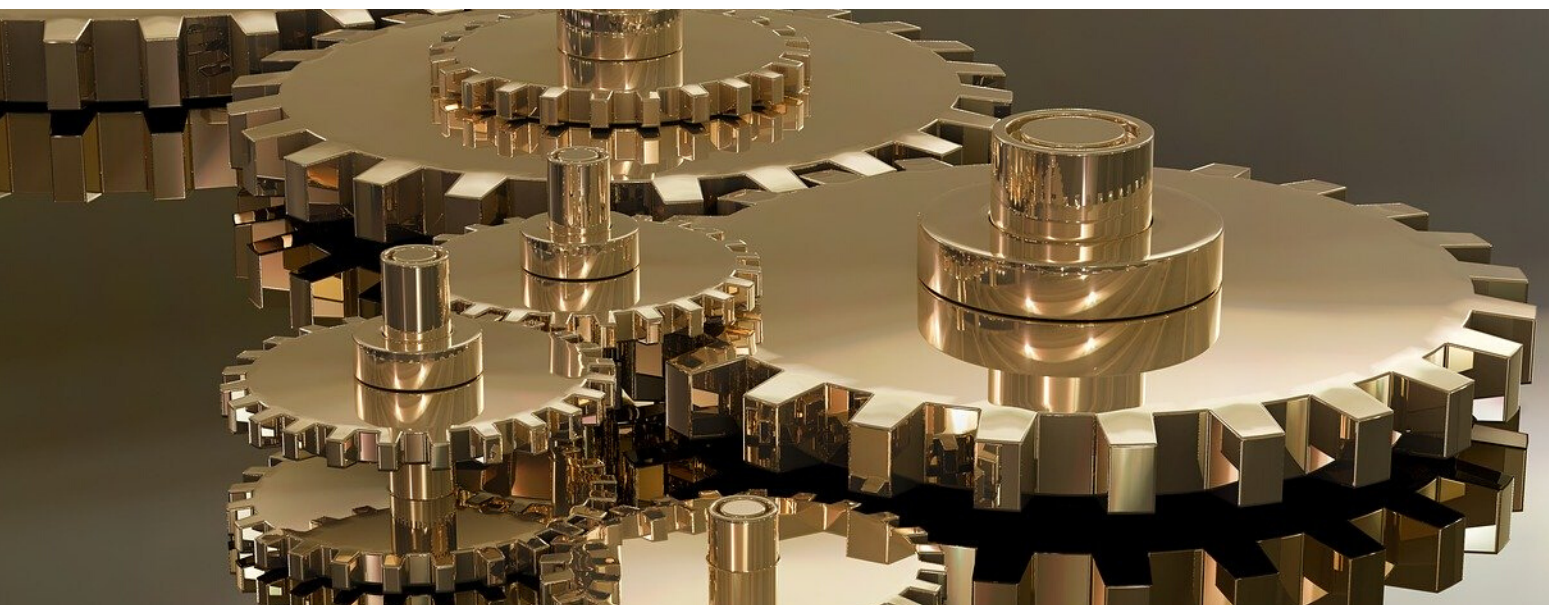
To minimise these risks, organisations using RPA should:

Establish the management of the cost of software quality as a formal project goal with

- **Stakeholder** = Funder;
- **Function** = Cost of Quality;
- **Attribute** = Minimum

Integrate the chosen **software engineering approach** into the project plan to **minimise the risks** inherent in the nature of the solution and in the RPA tools.

Integrate **software engineering methodologies** in a way that **takes into consideration the capabilities required** of the solution and the RPA tools to be used.



# USE IT RIGHT

“Using it Right” is not just a “how to” exercise with regard to the operational use of the RPA solution

Introducing RPA will have a significant affect on the organisation and on staff whose activities are being augmented or supported by the RPA solution.

Of course, the RPA solution itself will need to be carefully managed.

Key concepts that must be considered in an RPA project:

- **Organisational Change Management**
- **Activity Change Management**
- **RPA Management**



## USE IT RIGHT!

# ORGANISATIONAL CHANGE MANAGEMENT

Whether a single 'bot or an CRM suite, the introduction of a new technology solution will cause changes in how individual(s), team(s), function(s), or the whole organisation, operates.

The key is to identify all stakeholders.

Too often, too much stakeholder management is focussed on managers who interact with the project team and too little is spent on workers who will be directly affected by the change.

Consideration must also be given to how process knowledge, provided by employees and embedded in the software (robot) is to be maintained once the solution has been implemented.



### TIP

*Organisational Change Management must consider not only how to successfully deploy the solution, but also whether the organisational benefits are completely and correctly defined and are “real”.*

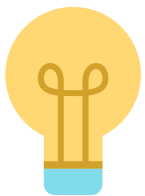
# USE IT RIGHT!

## ACTIVITY CHANGE MANAGEMENT

The majority of enterprise software deployments are made on the basis that the benefits are savings, particularly of “headcount”. The headcount that is “freed” by such projects present opportunities that are rarely seized. For example, the “best” employee may be an excellent candidate to become a key member of the IT/RPA organisation.

Most organisations have customer satisfaction levels below, sometimes very far below, 80%. This level of dissatisfaction reflects a significant waste of time, money and people’s capabilities. Unfortunately, RPA solutions designed to deliver “savings” do not do anything to reduce this waste, they simply deal with its consequences more “efficiently”.

Activity Change Management should **look at how this opportunity raised by the application/RPA project may be seized more broadly** than only within the affected team or operation.



### TIP

*Avoid spruiking the benefits of RPA and new applications as being “headcount reduction”. Instead focus on quality, speed and quality/compliance.*



# USE IT RIGHT!

## RPA MANAGEMENT

Once the RPA solution is deployed, the fun of its development and implementation is over and the benefits banked. Now you are in the boring time of keeping the 'bot or application ticking over.

Welcome to the world of IT Operations!

Operating and maintaining enterprise software is no trivial task – that's why there are IT departments and IT outsourcing providers.

The “Operator” and “Maintainer” stakeholders are critical to the continuing realisation of the benefits of the solution and are usually found within the IT department.

**IT is a critical stakeholder.**



### TIP

*To maximise the benefits of your RPA initiative, involve IT as early as a critical stakeholder*

# ALL THE WAY WITH RPA

## [BETTER TECHNOLOGY FOR YOUR BUSINESS]



To reduce the risk inherent in building something new and unique, you must be disciplined in responding to 4 key questions:

- 1. Why build anything?**
- 2. What is the right thing to build?**
- 3. How do I build it right?**
- 4. How do I use it right?**

This paper addresses these questions in a manner that ensures you build what you need and that what you build meets those needs at an appropriate cost.

*Philip Radburn  
Managing Principal  
BusinessSPM Pty Limited  
(c) 2019*

*Images: Pixabay (Gerd Altmann, Open ClipArt Vectors, Arek Socha) Marco Lachmann-Anker), Gartner, Forrester, Forbes, BusinessSPM (Radburn)*