Genius LOGIX

FREE GUIDE

A Practical Guide to

Application Rationalization

Before Cloud Migration for Growing Businesses



A free guide from **Genius Logix**

First edition, November 2026

First edition, November 2026

© 2026 Genius Logix. All rights reserved.

This guide is provided for general information. It does not replace legal, compliance or financial advice. Please check any decision that affects data protection, certifications or contracts with a suitable adviser.

Genius Logix · geniuslogix.com · info@geniuslogix.com

Contents

	Foreword	4
1	What application rationalization is	5
2	Why rationalize before you migrate	7
3	The strategy: keep, retire, replace or migrate	9
4	The process, step by step	12
5	Best practices and common mistakes	15
6	Costs, tools and a readiness checklist	17
7	When to bring in a partner	20
	Glossary	21
	Sources	21

Foreword

> **Cloud migration plans usually begin with a list of servers and a deadline. They rarely begin with a simpler question: do we still need all of this?**

We wrote this guide because that question is easy to skip and expensive to leave until later. A business grows, tools pile up, and by the time a cloud project starts, the estate holds applications nobody owns, tools that overlap and licences that renew out of habit. Moving all of it unchanged is the quickest way to make a new environment as cluttered as the old one.

Application rationalization is the answer. It is a short, structured review that sorts every application into keep, retire, replace or migrate before any move begins. It does not need special software, a large team or a long project. It needs an inventory, a simple scoring method and the willingness to make decisions.

This guide gives you that method in plain English. It is written for people who run growing businesses, not for specialists, and any term you do not recognize is explained in the glossary at the end. Each chapter leaves you with something you can use, from a decision path to a readiness checklist.

You can run the whole process yourself. If you would rather have company, our team is happy to help, and Chapter 7 explains how we work. Either way, we hope the next cloud project you plan is a smaller, cheaper and safer one.

The Genius Logix team

CHAPTER 1

01

What application rationalization is

Application rationalization is the process of reviewing every application your business runs and deciding, one by one, whether to keep it, retire it, replace it or move it. The aim is a smaller, cheaper and safer set of applications that fits the way you work today, not the way you worked when each tool was bought.

Most growing businesses end up with more applications than they planned. A team buys a tool to solve a problem, someone builds a quick internal system, a merger brings in three more, and nobody goes back to ask whether the old ones are still needed. Each tool made sense at the time. Together they become a set of overlapping systems, contracts and integrations that few people fully understand.

Rationalization is the step where you stop and look at the whole set.

How it differs from similar terms

Three terms are often used as if they mean the same thing. They do not, and mixing them up leads to the wrong plan.

Term	What it answers	Typical output
Application rationalization	Which applications should we keep, retire, replace or move?	A decision for every application
Application portfolio management	How do we keep track of our applications over time?	An ongoing register, reviewed regularly
Application modernization	How do we upgrade an application we are keeping?	A rebuilt, re-platformed or refactored system

Rationalization comes first. It tells you which applications deserve the effort of modernization and which should simply be switched off. Portfolio management is how you keep the result up to date afterwards.

Three signs your business needs it

1. **Overlapping tools.** Two teams use different products for the same job, such as separate project trackers, chat apps or reporting tools, and each has its own cost and its own data.
2. **Applications nobody owns.** When something breaks, nobody knows who is responsible. These are often the riskiest systems in the business, because nobody is watching them.

3. **Licences paid for but unused.** Subscriptions renew every year while the number of people actually logging in has dropped, sometimes to zero.

If you recognize even one of these, you will almost certainly find savings and simplifications in the review.

A worked example

> *The following example is illustrative. It shows how the method works on a typical small manufacturer and is not a description of a specific client.*

Imagine a manufacturer with 40 staff. It runs an ERP system for stock and orders, an old quoting tool that a former employee built in a spreadsheet and database, and four separate spreadsheets that sales, production and finance each use to track the same jobs. The company is planning to move its ERP to the cloud.

A rationalization review would ask the same questions of each item:

- **The ERP system:** it is central to the business and still supported, so the decision is to keep it and migrate it, which becomes the main focus of the cloud project.
- **The quoting tool:** only one person understands it and it cannot be supported long term. The decision is to replace it, and quoting moves into the ERP.
- **The four spreadsheets:** they duplicate each other, so the decision is to retire three and keep one, fed from the ERP.

The result is that the cloud migration moves one system, not six, and the business ends up with fewer places where the same information can disagree. That is the benefit of looking first.

The next chapter explains why doing this review before the migration, not after it, is the cheaper path.

CHAPTER 2

Why rationalize before you migrate

The cheapest application to move to the cloud is the one you did not need to move. That is the whole argument for rationalizing first, and the rest of this chapter explains why it holds up.

Every application you move has a cost

Moving an application is not a simple copy. Someone has to assess it, plan the move, test it, fix what breaks and support it afterwards. That work takes the same effort whether the application is vital or forgotten, so every application you move without need uses time and budget that could have gone to something that matters.

The move also creates ongoing costs. Cloud services are generally charged by what you use, so each workload you carry over adds to the monthly bill, and it keeps adding to it for as long as it runs. An application nobody uses still costs money if it is running in the cloud.

There is a security cost too. Each application you run is another thing to patch, monitor and protect. If you move an old system that is no longer supported, you take its weaknesses with you, and you now have to secure them in a new environment.

Moving first and tidying later is harder to undo

Some businesses decide to migrate now and sort out the applications afterwards. It sounds practical, but it usually goes the wrong way for three reasons.

1. **The pressure moves.** Once the migration is finished, the deadline has passed and the budget has been spent, so a tidy-up project struggles to get attention.
2. **The systems become entangled.** After a move, applications are connected to each other in new ways and to the cloud services around them. Switching one off later means untangling those connections.
3. **You pay for both.** While you decide what to retire, you carry the cost of everything, in the old environment and in the new one.

By contrast, a review done before the move is a one-off exercise with a clear outcome: a shorter list of applications, each with a decision attached.

The legacy problem

Many growing businesses are moving to the cloud because of an old system. It may be an in-house tool that has outlived its builder, an ERP that is no longer supported or a database that only one person understands. The temptation is to pick it up and drop it into the cloud exactly as it is, which is

often called lift and shift.

That can be the right call for a stable, well-understood application. For an unreliable one, it simply moves the unreliability to a new address. Rationalization asks the harder question first: is this system worth keeping, or should it be replaced as part of the move?

If you are dealing with an ageing system, our guide to legacy system modernisation for UK businesses covers the options in more detail.

What you gain by looking first

A review before the migration gives you four things that are hard to get afterwards:

- **A smaller migration.** Fewer applications to move means a shorter, cheaper project and fewer things that can go wrong.
- **A clearer budget.** You know what you are moving and why, so the plan and the cost estimate rest on facts.
- **Savings from the start.** Retiring duplicate tools and unused licences reduces spend immediately, not after the project ends.
- **A better target.** You choose the right cloud setup for the applications you are keeping, instead of designing around everything you happen to own.

The next chapter sets out the decision framework that makes this practical.

FREE CONSULTATION

Not sure which of your applications are worth moving?

Talk to Genius Logix about a free review.

Contact us →

CHAPTER 3

The strategy: keep, retire, replace or migrate

Every application ends up with one of four decisions. A good strategy is a simple, repeatable way of reaching that decision for all of them, so the result does not depend on who happens to be in the room.

The four decisions

Decision	When it fits	What it means in practice
Keep	The application works, the business needs it and it is well supported	Leave it where it is, or include it in the migration unchanged
Retire	It duplicates another tool, is barely used or no longer serves a purpose	Switch it off, archive the data you must keep and cancel the contract
Replace	The function is needed but the tool is weak, unsupported or too costly	Move the work to a better product, often one you already own
Migrate	The application is needed and worth keeping, but belongs in the cloud	Move it, in one of the ways described below

The decisions are not equally common. In most reviews, a surprising amount of the list falls into retire or replace, and what is left is the shorter set of applications that genuinely need migrating.

Scoring each application

To reach a decision consistently, score every application from 1 to 5 on four factors. A simple spreadsheet is enough.

Factor	What to ask	A score of 5 means
Business value	How much does the business depend on this? What stops if it disappears?	Critical to revenue, delivery or compliance
Technical health	Is it supported, stable and secure? Can it be changed safely?	Modern, supported and easy to maintain
Cost	What does it cost each year in licences, hosting, support and staff time?	Low cost for what it delivers
Risk	What is the exposure to security gaps, compliance problems or one person leaving?	Low risk

A useful rule is to give the scores to the people closest to each application, then have a second person challenge them. Scores given by one person alone tend to flatter the tools that person uses.

Reading the scores

- **High value and good health:** keep it. If it will run well in the cloud, plan to migrate it.
- **High value but poor health:** this is the most important group. It is needed but fragile, so decide whether to modernize it or replace it before you move it.
- **Low value and high cost:** retire it first. These are your quickest savings.
- **Low value and low cost:** leave it for now, or retire it when convenient. It is not worth a project of its own.

A low risk score on a valuable application does not change the decision, but it does change the order. Fix the risky ones first, because they are the most likely to cause a problem during the migration.

How the decisions connect to migration options

Once an application is marked migrate, there are several ways to do it. They are often called the Rs of migration, and the names vary slightly between providers. The common ones are:

Option	What happens	Best for
Rehost (lift and shift)	The application moves as it is to cloud servers	Stable applications that need to move quickly
Replatform	The application moves with small changes, such as a managed database	Applications that gain from cloud services without a rebuild
Refactor or rebuild	The application is redesigned to use cloud features properly	Valuable applications that are limited by their design

The scoring helps here too. A high-value, healthy application is a candidate for rehosting or replatforming. A high-value, poor-health application often needs refactoring, or replacing, to be worth the move.

Putting it together

Figure 1 below shows the decision path for a single application, from the first question to the final choice. Use it as a quick check when the scores leave you unsure.

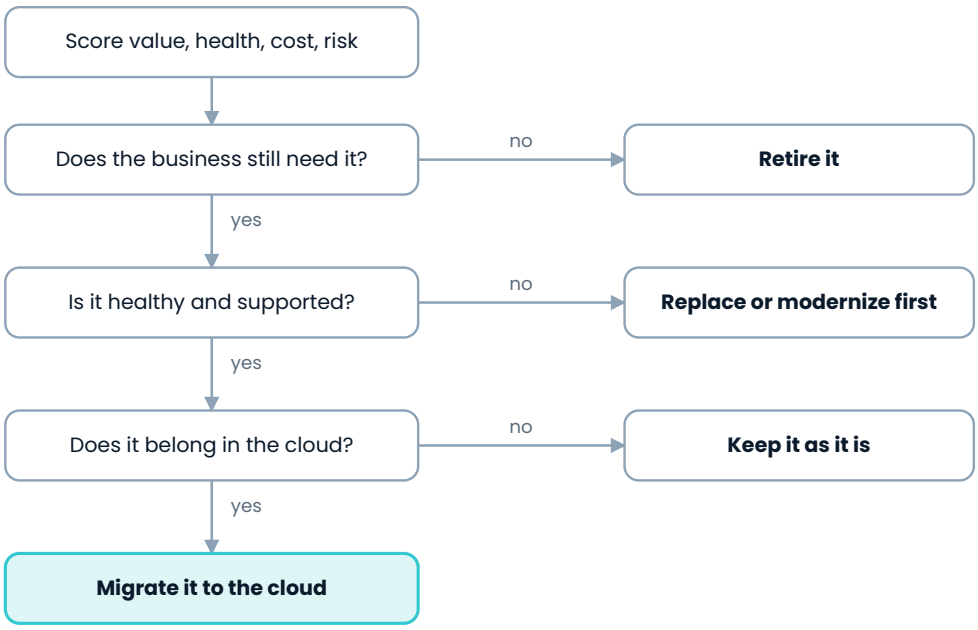


Figure 1 · Decision path for one application, 3 questions and 4 outcomes

Read it from the top. An application nobody needs is retired. One that is needed but fragile is replaced or modernized before anything moves. One that is healthy but does not belong in the cloud stays where it is, and the rest are migrated.

CHAPTER 4

The process, step by step

A rationalization review is a project with a start, a middle and an end. For a company with up to a few hundred staff, it usually takes four to six weeks of part-time effort from a small team. The six steps below are the same whether you do it yourself or with help.

Step 1. Build the inventory

You cannot make decisions about applications you have not listed, so the first job is a complete inventory. Start with what you know: the finance system, the CRM, the ERP, the email platform. Then widen the search, because the applications that cause trouble are often the ones nobody remembers.

Useful places to look are your payment records for subscriptions, your single sign-on or identity provider for what people log in to, your IT team's asset list and the contracts folder. Ask each department head what their team uses to do their job, including spreadsheets and tools bought on a company card.

Record the same details for every application, using a table like the one below.

Field	Example
Application name	Quoting tool
Business owner	Head of Sales
Technical owner	IT manager
What it does	Produces customer quotes
Number of users	6
Annual cost	Licence, hosting and support combined
Contract end date	The date it renews or expires
Where the data lives	On a local server
Connected to	ERP, email

The fields "business owner" and "connected to" matter most. The first tells you who to ask. The second tells you what else is affected when you change something.

Step 2. Score each application

Using the four factors from Chapter 3, give every application a score from 1 to 5 for business value, technical health, cost and risk. Ask the business owner for the first, the technical owner for the second and third, and have someone from IT or security give the fourth. Keep a short note beside each score explaining why, because you will need those reasons when someone challenges a decision.

Step 3. Make a decision for each one

With the scores in front of you, assign every application one of the four decisions: keep, retire, replace or migrate. Hold a short meeting for each business area, show the proposed decisions and agree them with the owners. This meeting matters as much as the scoring. An application can look worthless on paper and still be essential to one person's month-end routine, and that is the kind of thing only the owner will tell you.

Write down anything you decide not to act on yet, with the reason. A decision to leave something alone is still a decision.

Step 4. Sequence the moves into waves

Do not try to act on everything at once. Group the decisions into waves, ordered by effort and benefit.

- **Wave 1: quick wins.** Retire duplicate tools and cancel unused licences. These save money straight away and need little technical work.
- **Wave 2: replacements.** Move work from weak or unsupported tools to better ones. These need training and some data work.
- **Wave 3: migrations.** Move the applications that are staying, starting with the simplest and least critical, so the team learns on something low risk.

Be careful with contract dates. A licence that renews next month may be worth cancelling in wave 1, and one that has two years left can wait.

Step 5. Plan the migration for what remains

Only now do you plan the cloud migration, and it is a smaller project than the one you would have planned at the start. For each application marked migrate, choose the route (rehost, replatform or refactor), list its dependencies and agree a testing plan and a rollback plan. Decide how long the old and new versions will run side by side.

Step 6. Review after 90 days

Three months after the work is done, look again. Check that the retired applications are truly switched off, that the savings have appeared in the accounts and that nobody has quietly bought a replacement tool. Update the inventory and set a date to repeat the review. Once a year is a sensible rhythm for most businesses.

How long each step takes

Step	Typical effort for a growing business
1. Inventory	One to two weeks
2. Scoring	One week
3. Decisions	One week
4. Waves	A few days
5. Migration plan	One to two weeks
6. Review	A half-day, 90 days later

These are planning estimates, not guarantees. The biggest factor is how well you know your own systems, and a messy estate takes longer.

FREE CONSULTATION

Want help running the inventory and scoring?

We can do it with your team in the first 30 days.

Contact us →

CHAPTER 5

Best practices and common mistakes

Most problems in a rationalization project are not technical. They come from unclear ownership, rushed decisions and data that was not checked. This chapter lists what tends to work, what tends to go wrong and a short check to run before you switch anything off.

Best practices

1. **Name one owner for every application.** A shared owner is the same as no owner. Each application needs one person who can say whether it is still needed and who is responsible when it changes.
2. **Involve the people who use it.** The business owner knows how the application is really used, and often uses it in ways IT has never seen. Talk to them before you decide, not after.
3. **Work in waves.** Retiring a duplicate tool is a small, safe change. Replacing the system that runs your invoicing is not. Starting with the small ones builds confidence and savings before you take on the harder work.
4. **Keep the scoring simple.** Four factors scored from 1 to 5 is enough. A complicated model takes weeks to build and still depends on someone's judgement at the end.
5. **Record every decision and the reason.** Six months later, someone will ask why a tool was retired. A one-line note saves a long discussion.
6. **Tell people early.** Staff who learn on Monday that their tool is gone on Friday will find ways around the change. Give notice, training and a date.

Common mistakes

- **Scoring on technical grounds alone.** An old application can be completely sound for the job it does, and a modern one can be a poor fit. Business value must count as much as technology.
- **Skipping the data.** Retiring an application is not just switching it off. You may need to keep its records for tax, legal or customer reasons, and you need somewhere to put them first.
- **Retiring before the replacement is ready.** The gap between switching off the old tool and getting the new one working is where work stops and mistakes happen. Run them side by side for a short period.
- **Forgetting the connections.** An application that looks isolated may feed a report, an invoice run or another system. The "connected to" field in the inventory exists to catch this.
- **Treating it as an IT project.** If only IT is in the room, the review will miss how the business actually works. Include finance, operations and the people who use the tools.

- **Doing it once and stopping.** New tools appear every month. Without a regular review, the estate will grow back.

A note for UK businesses

Three points deserve a check before any application moves or is retired.

- **Where the data will live.** Personal data is covered by data protection law wherever it is stored. Check where a cloud provider will store and process your data, and whether your contracts and obligations allow it.
- **What a cloud move does to your security scope.** If you hold, or plan to apply for, Cyber Essentials, moving systems to the cloud can change what falls inside the certificate. Check this before the move, not afterwards.
- **Official guidance.** The National Cyber Security Centre publishes practical cloud security guidance that is worth reading before you choose a provider.

Five questions to ask before you retire anything

1. Who owns this application, and have they agreed to retire it?
2. What data does it hold, and where will we keep the records we must retain?
3. What other systems, reports or processes depend on it?
4. Is the replacement in place, tested and understood by the people who will use it?
5. Can we switch it back on if something goes wrong, and for how long?

If you cannot answer all five, the application is not ready to be retired. That is a useful result in itself, because it tells you what to do next.

CHAPTER 6

Costs, tools and a readiness checklist

This chapter covers three practical questions: where the money goes, what tools help and how to tell whether you are ready to start moving applications.

Where the money goes

A rationalization and migration project has two kinds of cost, the cost of doing the work and the cost of running the result. It helps to see both before you start, because the first is a one-off and the second continues every month.

Cost area	What it includes	One-off or ongoing
Review and planning	Staff time for the inventory, scoring and decision meetings, or a partner's fees	One-off
Migration work	Moving, testing and fixing applications, plus any rebuild or replatform work	One-off
Replacement tools	Licences, set-up and training for any new product that replaces an old one	Both
Cloud running costs	Compute, storage, network traffic and managed services, usually billed by use	Ongoing
Support and management	Monitoring, patching, backups and help for users	Ongoing
Parallel running	Paying for old and new systems at the same time during the changeover	Short-term

The saving from rationalization comes from the second half of that table. Every application you retire removes a licence, a share of hosting and some support work from the monthly total.

We have not put price ranges in this guide, because costs depend heavily on the size and condition of your estate, and a figure that is wrong for your business is worse than none. When you build your own estimate, work through each row of the table and ask the application owner for the real numbers.

Tools that help

You do not need specialist software to run a rationalization review for a small or mid-sized estate. Choose tools by the job they do.

Job	Tool type	Notes
Keep the inventory	A shared spreadsheet or a simple configuration database	Start with a spreadsheet and move later if it grows too large
Find forgotten applications	Payment records, your identity provider's app list and network discovery	Subscription and login records catch what IT never installed
Map dependencies	Discovery tools and interviews with application owners	Interviews catch what automated tools miss
Assess cloud readiness	The assessment tools from AWS, Azure and Google Cloud	Each provider offers its own, for example Azure Migrate, so check current documentation
Track the project	The task tool your team already uses	Do not buy another tool to manage the review

None of these tools makes the decisions. They give you better information, and the scoring from Chapter 3 turns that information into decisions.

The 15-point readiness checklist

Work through these points before you start moving anything. They are grouped by area. If you cannot tick a box, treat it as the next task, not a reason to stop.

PEOPLE AND OWNERSHIP

- [ ] A named sponsor in the leadership team supports the project.
- [ ] Every application has one named business owner.
- [ ] Someone has time set aside to run the review.

APPLICATIONS AND DATA

- [ ] You have a complete inventory, including spreadsheets and tools bought on company cards.
- [ ] Every application has a score for value, health, cost and risk.
- [ ] You know which data must be kept, for how long and where it will live.
- [ ] You have listed the connections between applications.

CLOUD READINESS

- [ ] You have a decision for each application: keep, retire, replace or migrate.
- [ ] Migration candidates have a chosen route: rehost, replatform or refactor.
- [ ] You have a sequence of waves and a date for each.
- [ ] Testing and rollback plans exist for the first wave.

SECURITY AND COMPLIANCE

- [ ] You know which applications hold personal or sensitive data.
- [ ] You have checked where a provider will store and process that data.
- [ ] Access controls, backups and monitoring are planned for the new environment.
- [ ] Someone has checked how the move affects any certifications you hold.

As a rule of thumb, if you can tick most of them, you are in a good position to start. If you can tick fewer than half, spend another week or two on steps 1 to 3 of the process before you plan the migration.

FREE CONSULTATION

Finished the checklist?

Send us your results and we'll tell you where to start.

Contact us →

CHAPTER 7

When to bring in a partner

07

Many businesses can run a rationalization review on their own, especially when the estate is small and the team knows it well. Outside help is worth considering when one of three things is true.

Three situations where a partner pays for itself

1. **Nobody has the time or ownership to do it.** The review needs someone to run it from start to finish. If your IT team is already full with day-to-day work, the review slips, and so does the migration.
2. **There is a hard deadline.** A data centre contract ending, a system going out of support or a growth target can leave no room for a slow review. A partner brings a tested method and the capacity to move quickly.
3. **A legacy system nobody fully understands.** When the person who built an application has left, or the documentation never existed, an experienced team can read the system, map its connections and recommend a safe route.

What a good partner does in the first 30 days

Whoever you choose, you should expect a clear, short start. In the first month, a good partner should:

- build the inventory with your team, not hand you a template and leave;
- run the scoring session with the business owners and challenge the scores;
- give you a written plan with decisions, waves and the effort for each wave;
- tell you honestly which applications you do not need to move, even when that reduces the size of the project.

The last point is a good test. A partner who only ever recommends moving more is not working for you.

How Genius Logix works

Genius Logix is a UK software and IT consulting firm founded in 2023. We help businesses choose, build, connect, move and run their software, across custom development, ERP integration, cloud migration and ongoing support.

A conversation with us starts with a free consultation, where we listen to what you are trying to do and tell you whether we can help. If we can, we follow with a written proposal. We work as one team across the stages, from strategy and selection through implementation and integration to managed services, so the people who planned the work stay involved after it goes live. You can see how that looks on a real project in our cloud migration case study.

Glossary

Term	Meaning
Application rationalization	Reviewing every application and deciding whether to keep, retire, replace or migrate it
Rehost	Moving an application to the cloud as it is, often called lift and shift
Replatform	Moving an application with small changes so it can use cloud services such as a managed database
Refactor	Redesigning an application so it takes full advantage of the cloud
CMDB	Configuration management database, a record of your systems and how they connect
TCO	Total cost of ownership, the full cost of an application over its life, not only the licence
Workload	An application or process running on a server or in the cloud
Dependency	Another system, report or process that relies on an application
Legacy system	An older application that is still in use, often hard to support or change
Wave	A group of applications that are changed together as one stage of the project

Sources

- National Cyber Security Centre, Cloud security guidance
- National Cyber Security Centre, Cyber Essentials overview
- Genius Logix, Legacy system modernisation for UK businesses

Genius LOGIX

Ready to decide what stays, goes and moves?

Book a free consultation

geniuslogix.com/contact-us

info@geniuslogix.com

© 2026 Genius Logix. First edition, November 2026.