Breaking the Shackles of Legacy Systems

A step-by-step guide to identify which applications to modernize and the best modernization approach

Neil Hartley, Morphis Technologies
Phone: 720 289 4403
e-mail: neil.hartley@morphis-tech.com
Most businesses can’t innovate sufficiently to maintain their competitive edge because of the spiderweb of architectures and technologies they’ve built up over years…

75% of North American and European enterprise IT budgets are expended on ongoing operations and maintenance.

90% of IT decision makers claimed that legacy systems are preventing them from harnessing the digital technologies they need to grow and become more efficient.
Within the spiders web, many businesses don’t even have a clear view of their application landscape…

“Many companies don’t know what applications they have, don’t know the interdependencies, don’t know if they’re sharing data models”
…Cameron Jenkins, Dell Services.

Let alone what these applications do…

“A lack of knowledge of the current system’s full capabilities is the biggest reason for project failure” - from ‘How Application Service Providers Can Help In Modernization Initiatives’ by Dale Vecchio, Gartner.
And then there is the question of trusted advisors to your business.

Is your preferred system integrator going to provide the best advice to you if it is going to impact their business negatively?

For example, if your SI has a healthy revenue stream from maintaining legacy applications on the mainframe, are they really going to advise you to replace/modernize/obsolete those applications?

And, of course, the vast majority of vendors are from the Abraham Maslow school of “if you only have a hammer, you tend to see every problem as a nail”.

Breaking the Shackles of Legacy Systems

Can You Trust Your Trusted Advisors?
The intent of this document is to introduce the reader to a framework that can be used independently to increase business innovation by answering the following questions:

- What applications do I have and what do they do?
- How important are they to my business?
- How do I rank them in terms of cost/risk/value?
- How do I determine what investment approach to take with each application?

So let’s dive in and liberate business innovation from the shackles of legacy systems…
Inability to innovate is the #1 driver for legacy modernization.

Source: Survey at the Gartner AADI Summit (Dec 1-3, 2015, Las Vegas)
Breaking the Shackles of Legacy Systems

Lack of Business Agility

Lack of business agility refers to the business’s inability to innovate to meet changing market requirements/threats. Examples include:

- Improved customer experience, e.g. customer demands for mobile access \( (43\% \text{ of IT executives believe an inability to work with legacy systems is the biggest barrier to future mobile initiatives...CIO Insight}) \).

- Threats from new market entrants (\textit{witness Workday’s entry to the Student Information System higher education market}).

- Security threats (\textit{the OPM blamed their inability to encrypt decades old legacy systems for recent data breaches}).

The need for businesses to innovate is increasing dramatically but IT’s flexibility and ability to respond is diminishing, largely because of the drain on budget and resources caused by legacy systems.
Of course, not every innovation in a business requires a technology solution, nor does every technology solution address a single business problem only.

**Example** - a fraud detection solution at First Notification Of Loss (FNOL) in an insurance business will not only score potential fraudulent claims immediately, it will also score good claims at FNOL allowing immediate settlement and driving customer satisfaction higher.

That said, let’s discuss the application landscape and how to map it in terms of cost/risk/value to the business…
The best “independent” approach we have seen to doing this comes from Gartner with their Pace Layers and TIME model.

Gartner’s Pace-Layered Application Strategy is a methodology for categorizing, selecting, managing and governing applications to support business change, differentiation and innovation.
Pace Layers is essentially a way to categorize applications in the business by their contribution to the business:

- **Systems of Record** - an application that performs a function where there is no value to the business in doing things differently to the competition. Examples might be HR, AR etc. These applications change little and functionality is often offered by COTS (Commercial-Off-The-Shelf) packages.

- **Systems of Differentiation** - an application that supports the differentiation of the business in its market. These applications tend to change more frequently and tend to be custom apps rather than COTS.

- **Systems of Innovation** - applications that are addressing new ideas and are under evaluation to see if they merit further investment.

Following is an example to illustrate how this categorization may work across a business...

---

**Breaking the Shackles of Legacy Systems**

*Gartner’s Pace-Layered Approach*
Breaking the Shackles of Legacy Systems

Use Pace Layers to Logically Decompose the Monolith
Not every business would characterize these same applications in exactly the same way but this clearly illustrates the first step in mapping out and categorizing applications by their contribution to the business.

Note that Gartner has toolkits to help the business apply the Pace-Layered approach.

They also provide scoring mechanisms to further categorize each application by its cost/risk/value to the business which then enables decision-making on the best course of action for each application.

Welcome to the TIME model…
Breaking the Shackles of Legacy Systems

Gartner’s TIME Model

- **Technical Condition** rates the quality of the application and takes into account the cost and risk associated with each application.
- **Business Value** simply rates apps by their value to the business.
Applications that fall in the **Tolerate** quadrant have relatively low business value but are in good condition and require low maintenance. All things considered these applications can be tolerated.
Applications that fall in the **Invest** quadrant are low maintenance, have high business value and, therefore, are worthy of continued investment.
Breaking the Shackles of Legacy Systems

**Gartner’s TIME Model**

- Low quality apps (a poor combination of quality/cost/risk) that provide high business value require “migration”, i.e. they need to moved out of this quadrant.
- See the next section for “migration” strategies based on the the driver (the primary reason for low quality) and the apps’ categorization within the Pace-Layered model.
Breaking the Shackles of Legacy Systems

Gartner’s TIME Model

- Those applications that have little business value and are in poor technical condition need to be **eliminated**.
So you have applications that fall into the “Migrate” quadrant of the Gartner TIME model and you need to move them to another quadrant. Which “migration” approach is best?

There are a number of variables that influence this decision including:

1. **Which PACE Layer** does the application belong to - Systems of Record or Systems of Differentiation?

2. **What is the primary driver to migrate** (i.e. the primary reason the app is categorized as “low quality”)?
Breaking the Shackles of Legacy Systems

*Legacy System “Migration” Alternatives*

“Low quality” issues can include:

- **UX** - the app does not support the channels and use models demanded by customers/users (e.g. mobile, online, cloud).

- **Functionality** - the app has lacked the investment required to keep up with user/customer demands

- **Cost** - the app has become expensive to maintain, perhaps because it is built on an outdated technology stack

- **Security** - dependence on an outdated stack has introduced security concerns

- **Skills** - there is business risk associated with a shortage of skills to support and develop the app
Other variables that influence the choice of “migration” approach:

3. What are the costs of the “migration” option?

4. What are the associated risks?

5. How long will it take (time-to-value)?

6. What competitive advantage will this approach yield? If a “migration” option only provides a low competitive advantage then this approach would not be applicable for apps categorized as Systems of Differentiation.

Nomenclature lacks consistency in the legacy/application migration/modernization space so we will try to bring some order to that chaos as we discuss the options and their suitability for the various use cases…
Breaking the Shackles of Legacy Systems

Legacy System “Migration” Alternatives

Often thrown into the pot of modernization strategies is **Consolidation**. Can the functionality of the application be absorbed by another app? This is more likely to be for an app in the E quadrant and categorized as a Systems of Record but we capture it here for completion.

The next two options we will only touch on briefly as well.

- **Re-platforming** - refers to taking existing code and moving it from one platform (e.g. mainframe) to another (e.g. client-server). The code is not modified in any way. This is only applicable where reducing cost is the number one driver. If other quality drivers are, or become, an issue then re-platforming will not help and can only be used as a stepping stone to re-architecting the application. Often referred to simply as “Migration”.

- **Refactoring** - refers to taking an existing code base and, usually with the help of technology, identifying duplicate and redundant code so as to simplify the code base and achieve reduced maintenance and hosting costs. Again, if other quality drivers are, or become, an issue then refactoring will not help. Also referred to as remediation.
There are then 4 major alternatives to address applications in need of “migration”:

- **Package** - the application is replaced by a COTS solution. This may also see a move from an on-premise app to a SaaS solution. We have seen this referred to as Migration too.

- **Rewrite** - the application is rewritten from scratch in the code of the new target environment.

- **Wrapping** - take the application as it is and apply a connectivity layer that will enable web/mobile deployment and improve user experience. Sometimes referred to as non-invasive application modernization.

- **Re-architect** - this approach takes the legacy application in whatever language it is in and transforms that code to a new target environment, typically a multi-tier architecture. Achieved using a combination of technology and services. Sometimes referred to as invasive application modernization.
Breaking the Shackles of Legacy Systems

The 4 Main “Migration” Alternatives

In terms of the basic comparators of “migration” cost/risk/time-to-value/competitive advantage, here are how the 4 approaches compare:

<table>
<thead>
<tr>
<th></th>
<th>Migration Cost</th>
<th>Migration Risk</th>
<th>Time-to-Value</th>
<th>Competitive Advantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>High</td>
<td>Medium</td>
<td>Mid/Low</td>
<td>Low</td>
</tr>
<tr>
<td>Rewrite</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Wrapping</td>
<td>Lowest</td>
<td>Lowest</td>
<td>Lowest</td>
<td>Mid/Low</td>
</tr>
<tr>
<td>Re-architect</td>
<td>Low</td>
<td>Low</td>
<td>Low/Mid</td>
<td>High</td>
</tr>
</tbody>
</table>

Clearly for those applications that fit into Systems of Differentiation, the **Package approach is not suitable** given its low competitive advantage.

Wrapping is a possibility for a Systems of Differentiation application but only as a stepping stone to a rewrite or a re-architecting. For continued functionality enhancements and contribution to the differentiation of the business, **Wrapping is not a good solution**.
Breaking the Shackles of Legacy Systems

The 4 Main “Migration” Alternatives

Here is how each “migration” alternative maps against the drivers for modernization, i.e. the causes of low quality in the Gartner TIME model.

<table>
<thead>
<tr>
<th></th>
<th>UX</th>
<th>Functionality</th>
<th>Cost</th>
<th>Security</th>
<th>Skills</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td>Depends</td>
<td>Poor</td>
<td>Poor</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Rewrite</td>
<td>Good</td>
<td>Good</td>
<td>Medium</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>Wrapping</td>
<td>Good</td>
<td>Doesn’t help</td>
<td>Doesn’t help</td>
<td>Good</td>
<td>Doesn’t help</td>
</tr>
<tr>
<td>Re-architect</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
</tr>
</tbody>
</table>

Hopefully this framework provides useful pointers when navigating through the maze of application modernization alternatives.

One aspect not touched upon yet is how to gain a full understanding of what each application does. As was pointed out earlier, “a lack of knowledge of the current system’s full capabilities is the biggest reason for project failure”…
Breaking the Shackles of Legacy Systems

Application Analysis

**Morphis Docmine** is a world class solution for legacy source code mining. Enabling an understanding of legacy code through GUIs and documentation, **Morphis Docmine** will reduce cost by identifying redundant/duplicate code and determine code complexity by dynamically establishing code relations and structure.

Available today as a service, **Morphis Docmine** can mine several programming languages and is the foundation for accurately assessing the cost and risk of re-architecting applications.

**Morphis Docmine** will be available as a SaaS solution from Q2 2016. If you are interested in signing up as a beta customer or would like more information please click [here](#).
Morphis Technologies

Neil Hartley
neil.hartley@morphis-tech.com
www.morphis-insights.com
720 289 4403