

# Texas Dept. of Information Resources Legacy System Study (LSS)

---

## Preliminary Analysis and Recommendations Report



June 9, 2014

Engagement: 330020788

## Table of Contents

<b>Overview and Objectives</b> .....	<b>1</b>
<b>Determining Software and Hardware Legacy Status</b> .....	<b>2</b>
Normalization of software products and versions.....	3
Determination of software support status .....	4
Determination of platform viability.....	7
<b>Defining Remediation Categories</b> .....	<b>9</b>
Remediation Categories .....	11
Legacy application categorization and remediation options .....	13
<b>Next Steps</b> .....	<b>15</b>
<b>Appendix A: Analysis Metrics</b> .....	<b>17</b>
<b>Appendix B: Gartner Market Clocks</b> .....	<b>18</b>

## Overview and Objectives

The Preliminary Analysis deliverable captures the approach and results of *Task 2 Assess Current State* of the Legacy System Study (LSS).

Objectives of the current state assessment are:

- Determine the legacy status 3rd party software products that are used by agency business applications
  - Using data supplied by agencies and from DCS' Configuration Management Database (CMDB)
  - Based on vendor end-of-life, end-of-support and end of extended support dates
- Determine the legacy status of hardware that supports Agency business applications
  - Using data supplied by agencies and from DCS' Configuration Management Database (CMDB)
  - Based on Gartner's research of the viability of server and operating environments
- Develop a legacy categorization framework taxonomy
  - Describe the characteristics of legacy applications
  - Helps map remediation of legacy systems to candidate remediation options
- Develop categories of remediation options
  - Identify the range of remediation options for business applications that are considered legacy
  - Describe categories of options

This document is structured into three parts:

1. Approach to determination of the software and hardware legacy status
2. Initial remediation categorization
3. Overview of the next steps in the analysis

## Determining Software and Hardware Legacy Status

Business applications are implemented using a number of technology components. Collectively, these components are referred to as the technology *stack*, indicating that components operate at different levels.

The technology stack consists of:

- Enabling software, such as application server and databases
- Supporting software, such as backup/restore and monitoring
- Operating environments, such as operating systems and virtualization
- Server technology, such as the hardware platform

For business applications in Texas agencies' portfolio, the underlying software and hardware components are products, whether commercial off-the-shelf (COTS) products, or products created by an open source community. Even when agencies develop their own business applications, the technical underpinnings of such applications use software products.

The component product lifecycle phase is a driver for the legacy status of business applications: when the components have reached legacy status, the business application has reached legacy status as well. In many cases, the remediation may be as straightforward as an upgrade to the current version of a product, in other cases, the remediation may be more complicated.

The legacy driver for enabling and supporting software is whether the software is still actively supported by the software vendor, or by the open source project community in case of open source products.

The legacy driver for operating environments and server technology are the Gartner Market Clocks that cover these respective tiers in the technology stack.

The following diagram illustrates the technology stack and legacy drivers:

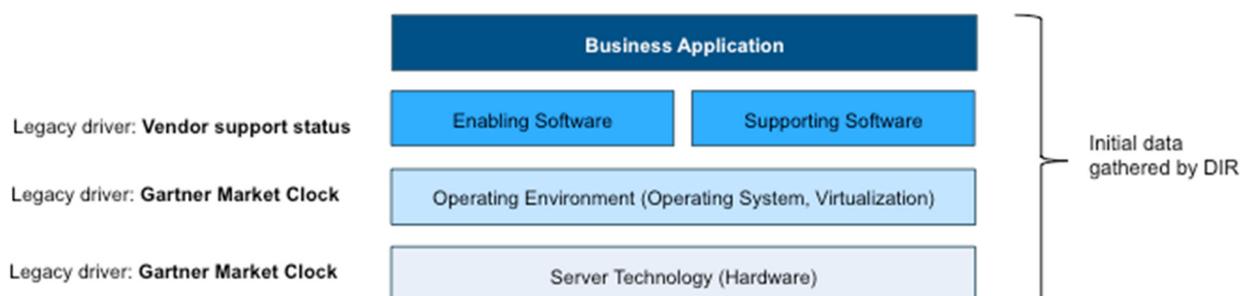


Figure 1 - Business application technology stack and legacy drivers

## Normalization of software products and versions

Agencies use many business applications to support their operations, which in turn use many software products. A cumulative list of these products exceeds well over 10,000 items. However, many of the software products are not unique to business applications, and not unique to agencies. For example, a permitting application in one agency could use the same database management software as the agency's records management application, or another agency's grant tracking application.

In order to reduce the list of software products to be evaluated for legacy status, we *normalize* the consolidated list of agency software products by grouping instances of software products under a normalized name.

The normalization process includes:

- Eliminating duplicates
- Fixing common data capture anomalies
  - Inconsistencies between vendor and product names
  - Anomalies introduced by automatic discovery tools
  - Data entry errors
- Grouping software version numbers to the level of granularity that aligns with the product support lifecycle
  - In the majority of cases, this consists of a major and minor version number (e.g. 2.1), or a major version only (e.g. 2)
  - In some cases, the version numbers are more granular (e.g. 2.1.1)
- In some cases, internal version numbers (such as build codes) or file versions must be translated to commonly understood version
  - For example, many Microsoft products were reported with build numbers (e.g. SQL Server 9.00.1399.06 instead of SQL Server 2005)
- Accommodating for vendor and software name changes over time, through acquisitions or branding
  - For example, Sun is now owned by Oracle

The process results the creation of two data sets:

- A normalized list of software
- A cross-reference between specific software used by business applications, and the normalized software

This process is partially automated, using pattern recognition, and supports incremental loads of agency data. A significant portion requires manual effort due to the variance of agency reporting.

The following diagram illustrates the high-level normalization process and its outputs:

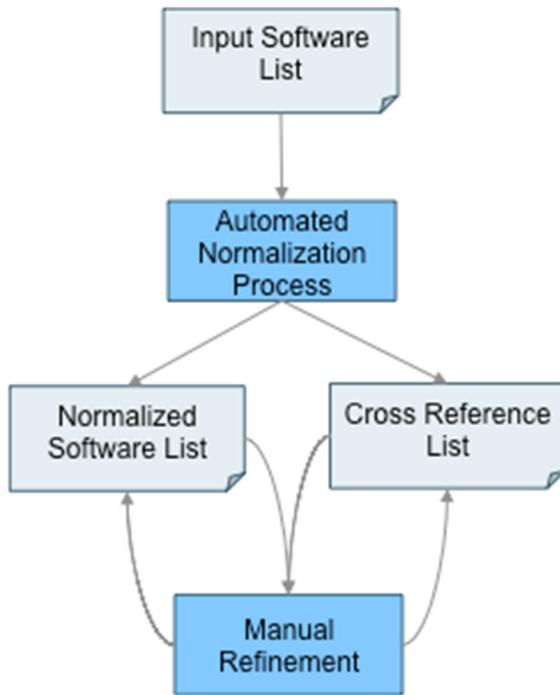


Figure 2 - High-level normalization process

## Determination of software support status

The software product support cycle is typically comprised of the following stages:

- Active support
  - Between the date when the project is released until some time after the product (or version) is withdrawn from the market, the vendor provides active support
  - Support typically entails bug fixes, security patches, and minor enhancements
- Extended Support (optional)
  - In some cases, vendors offer reduced support after active support window has closed
  - Support often entails critical bug fixes, access to the product knowledge base and problem resolution on a best effort basis
  - Extended support duration and services depend on the vendor and the product, and often requires additional fees
- No support
  - Product support has discontinued completely

For the Legacy System Study, products that are no longer under active support are deemed to be legacy products.

This timeline shows the typical software support lifecycle use by product vendors. Not all software vendors distinguish between these milestones:

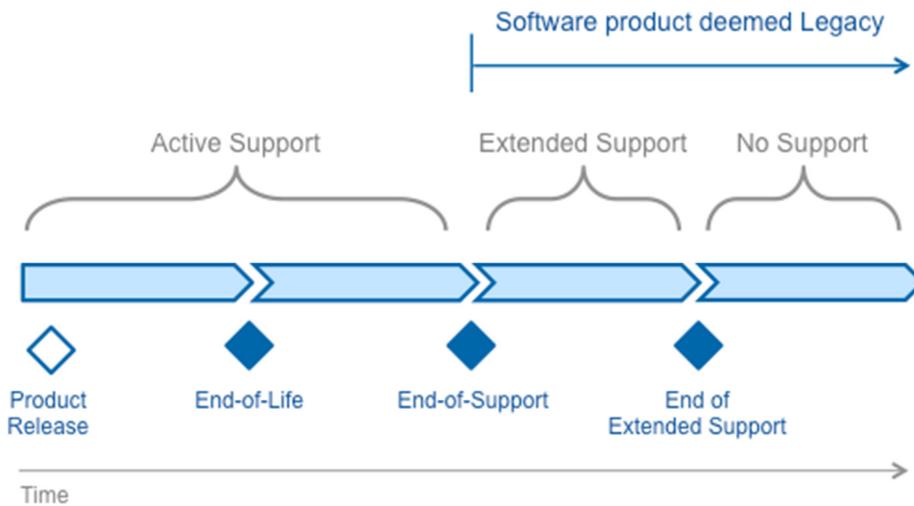


Figure 3 - Typical software support lifecycle used by product vendors

Open source software vendors use a similar lifecycle, with notable exceptions:

- The current and previous version of popular open source product are typically supported by the open source community
- At some point, the community leads decide to archive an older version of the product, and active support rapidly diminishes
- Non-mainstream open source products may never reach the required critical mass to be considered having community support

The following timeline illustrates the typical open source lifecycle:

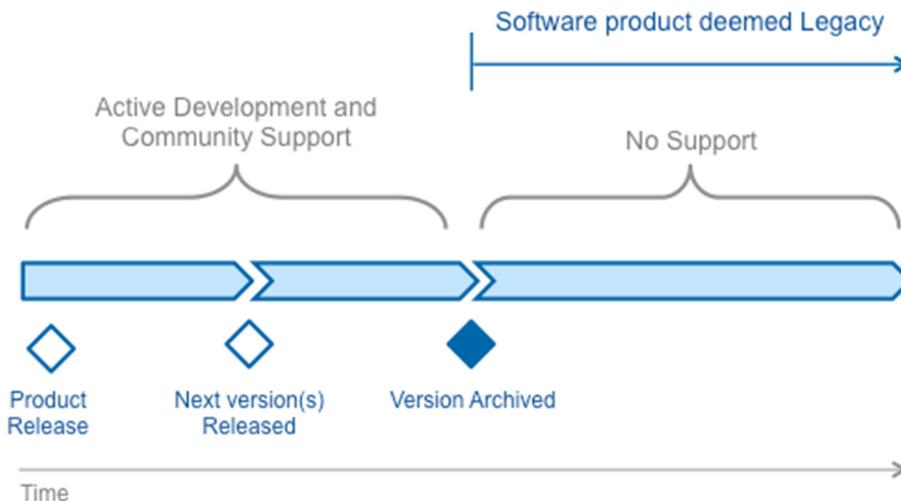


Figure 4 - Typical open source software lifecycle

For each product in the normalized software list, we attempt to capture the following data elements:

- **End-of-Life (EOL)** date, indicating when the vendor stopped selling the product
- **End-of-Support (EOS)** date, indicating when active support for the product is dropped or when an open source product version is archived
  - EOS is typically one or more years after EOL
  - For purposes of the Legacy System Study, a software product is considered legacy when the EOS date is past the cut-off date (8/31/2014)
  - For open source software products, EOL typically coincides with EOS, when the version archived
- **Extended Support** date, which sometimes provides limited support after the active support period has ended

In some cases, product lifecycle analysis results in updates to the software normalization. It may turn out that the support lifecycle is defined at a different level of granularity, or that products can be combined. This also results in updates to the cross reference list, which links agency supplied software entries to the normalized list.

The following diagram illustrates the high level process to determine the software legacy status:

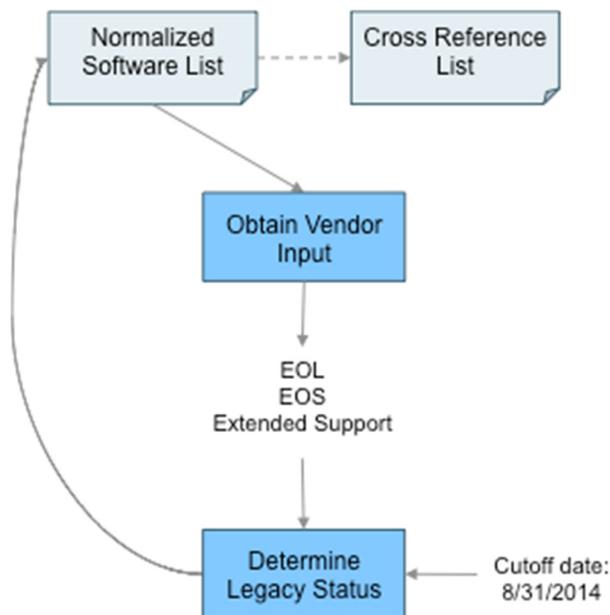


Figure 5 - Software legacy determination

## Determination of platform viability

Beyond the legacy status of the business applications' software components, the Legacy System Study considers the viability of the platform that these components are deployed to. The platform consists of:

- Server technology, or the hardware environment
- Operating environment, or the operating system and possibly virtualization

We use Gartner's IT Market Clocks for server technology and server virtualization/operating environments to determine the platform legacy status. The Gartner's IT Market Clocks is a decision framework that uses a clock-face metaphor to represent relative market time.

Technologies are positioned on the IT Market Clock using two parameters:

1. Where they currently lie within their own useful market life (from the first time the technology product or service can be acquired and used to the last time it can be viably used). This determines the rotational position of the asset on the Market Clock — each begins at 0 (called "Market Start"), and moves clockwise round to 12 o'clock □
2. Relative level of commoditization (the ease with which the technology product or service can be interchanged with alternatives). Relative commoditization determines the distance from the center of the Market Clock — assets further from the center are more commoditized. For most asset classes, relative commoditization levels begin low, increase steadily as the market matures and then decrease again toward end of life.

The IT Market Clock is divided into quarters, each of which represents one of these four market phases. The quarters are named to highlight the general approach recommended for technology passing through the corresponding market phase — advantage, choice, cost and replacement:

- **Advantage:** Assets in the customized phase, which provide differentiated technology, service or capability. There will usually be limited supply options and high dependence on relevant skills.
- **Choice:** Assets in the mass-customized phase, subject to increasing levels of standardization and growing supply options.
- **Cost:** Assets in the commoditized phase. Differentiation between alternative sources is at its minimum level and competition centers on price.
- **Replacement:** Assets in the disfavored phase, usually legacy technologies, services or capabilities. Supply choices and access to available skill sets will be decreasing, leading to rising operational costs. Their retirement or upgrade is essential.

For the Legacy System Study, we consider the technology to be legacy when it is in the replacement quarter.



Operating environments used by agency business systems include:

- Discontinued OSs such as Windows NT/2000/2003 and Netware
- Aging OSs that become less viable, such as HP/UX and z/VSE
- Modern Windows OSs, such as 2008 and 2012
- Viable UNIX variants, such as AIX and Solaris

The following diagram illustrates the overall grouping and legacy status:

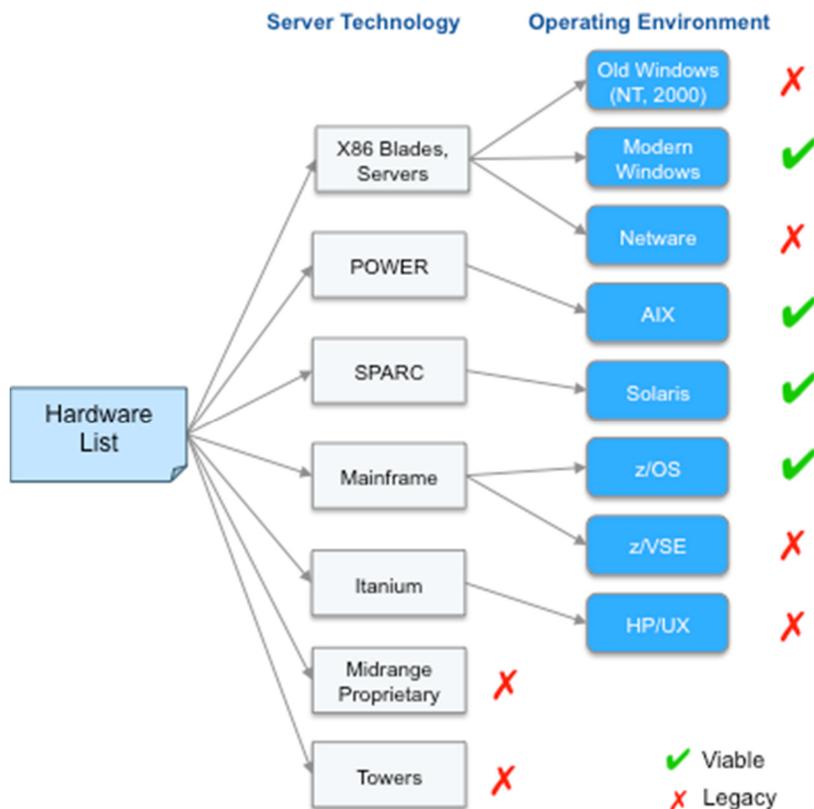


Figure 7 - Platform legacy status

## Defining Remediation Categories

Gartner uses the TIME model as a first level remediation categorization for rationalization of large application portfolios:

- Tolerate: Retain the business application without significant changes
- Invest: Address the shortcomings of the application by investing in upgrades or enhancements
- Migrate: Replacing the application's functionality with other existing or new applications

- Eliminate/Retire: Retire the business application, replacing its functionality with other existing or new applications

This high level model will be expanded in the next section.

There is no straightforward mapping between legacy business applications and remediation options. Beyond the technology drivers, potential remediation options depend on the application type, business value, ongoing cost and risk of the business application. The business value is assessed using a number of characteristics, including criticality, usability and productivity, which will be obtained through the online assessment that agencies plan to complete in June 2014.

The following diagram illustrates the influences to remediation options:

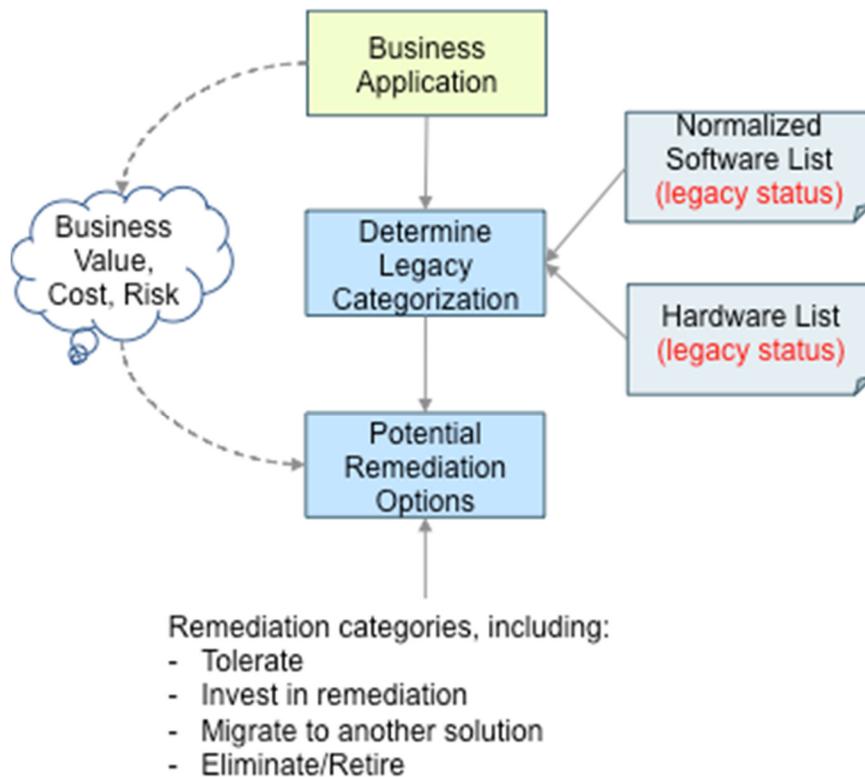


Figure 8 - Business application characteristics affect remediation options for legacy systems

## Remediation Categories

This section lists the legacy system remediation categories, expanded from the TIME model.

Remediation Category	Description	Possible Fit For
Eliminate/Retire	Retire the business application, using functionality of other applications to perform the business function	<ul style="list-style-type: none"> <li>Legacy applications that are low value, providing functionality that can be supported by other applications in the current portfolio</li> <li>Example: secondary financial systems that track expenditures that can be managed within the statewide financial system</li> </ul>
Tolerate/No Change	Leave the business application as-is	<ul style="list-style-type: none"> <li>Legacy applications that are high value, low cost, have a low degree of change and are relatively low risk</li> <li>Example: staff-facing asset management applications that run on older technology</li> </ul>
Replatform Hardware	Move the business application to a new server technology	<ul style="list-style-type: none"> <li>Legacy applications that are high value and have a sound software technology foundation, but run on aging hardware that can be replaced without significant impact to the business application</li> <li>Example: licensing systems that use modern enabling software on aging servers that can be ported to modern hardware</li> </ul>
Upgrade Software Stack	Upgrade underlying enabling software (application servers, databases, etc.) to current versions	<ul style="list-style-type: none"> <li>Legacy applications that are high value and have a sound architecture, but run on outdated software with manageable impact to the business application</li> <li>Example: permitting systems that use older versions of .NET environments or SQL Server databases that is superseded by modern versions</li> </ul>
Upgrade COTS	Upgrade the Commercial Off-the-Shelf (COTS) software, that implements the business application, to a current version	<ul style="list-style-type: none"> <li>Legacy applications that are high value and are implemented with a viable COTS solution, but runs on an old version of the COTS solution that can be upgraded with a manageable impact to the business application</li> <li>Example: human resource systems that use older versions of an ERP solution that is superseded by modern versions</li> </ul>

Remediation Category	Description	Possible Fit For
Transform Code	Transform the application code of a custom application from a legacy programming language/platform to a current platform, without making significant functional changes	<ul style="list-style-type: none"> <li>Legacy applications that are high value and generally meet the current and anticipated business needs, but have been built with a programming language or platform that can no longer be maintained by readily available resources in the marketplace</li> <li>Example: registration applications that run on COBOL and non-relational databases</li> </ul>
Consolidate	Combine the functionality of a legacy application with another existing application	<ul style="list-style-type: none"> <li>Legacy applications that provide important functionality that is similar to functionality provided by another existing system, which could feasibly incorporate this functionality</li> <li>Examples: <ul style="list-style-type: none"> <li>Provider management system that tracks vocational schools, which is similar to tracking adult education providers</li> <li>Agency financial systems that are similar to financial systems for other agencies</li> </ul> </li> </ul>
Replace with COTS (Software as a Service or on-premise)	Replace a custom or aging Commercial Off-the-Shelf (COTS) system with a modern COTS solution	<ul style="list-style-type: none"> <li>Legacy applications that support important business functions, but are implemented with aging custom developed software or obsolete COTS software that cannot be readily upgraded, but is readily supported with modern COTS solutions</li> <li>Examples: <ul style="list-style-type: none"> <li>Custom developed case management system that has a good functional fit with COTS case management solutions</li> <li>Aging Telligent content management system that no longer has an upgrade path</li> </ul> </li> </ul>
Rewrite	Replace the business application with a new custom developed solution	<ul style="list-style-type: none"> <li>Legacy applications that support important business critical functionality, but cannot be remediated with any other option</li> <li>Example: business application that supports unique Texas capability</li> </ul>

## Legacy application categorization and remediation options

The online assessment will collect additional application characteristics that help to recommend the appropriate remediation option within the categorization framework.

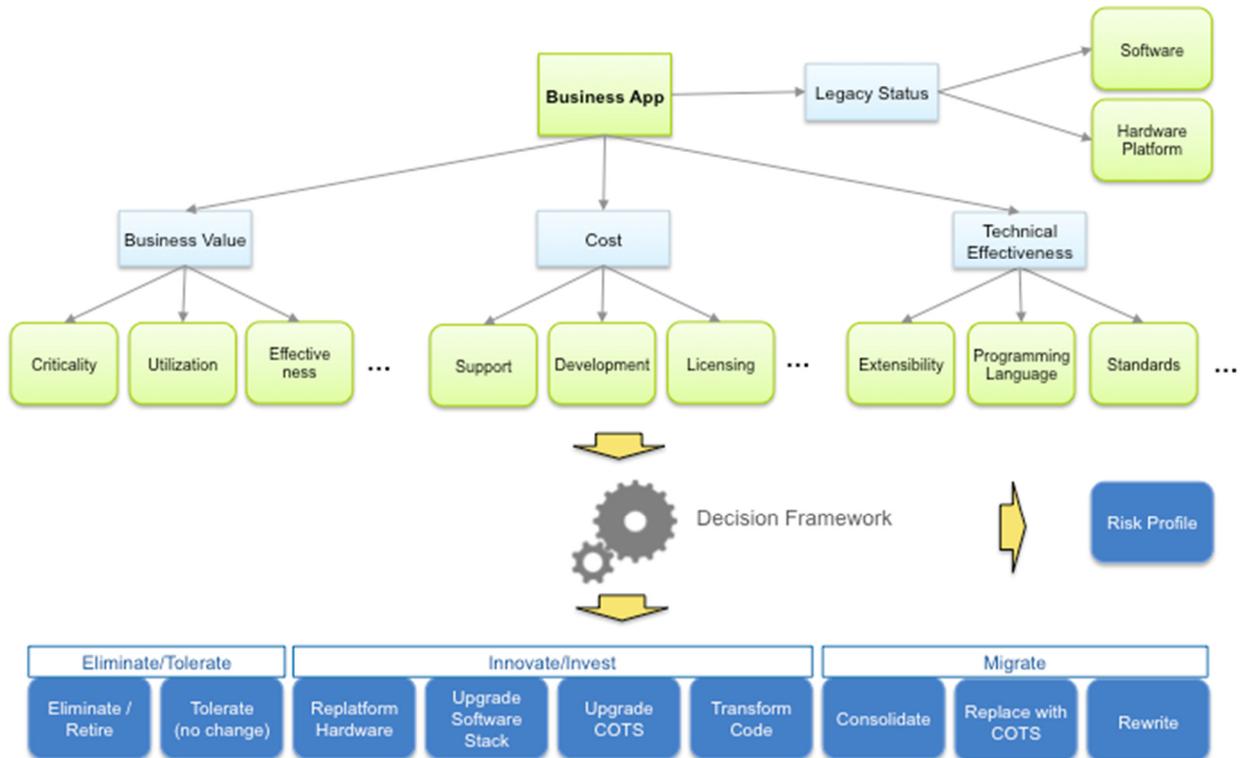


Figure 9 - Matching business application characteristics to remediation categories

The following diagram illustrates how the online assessment augments drivers for the top layer of the technology stack, the business application itself:

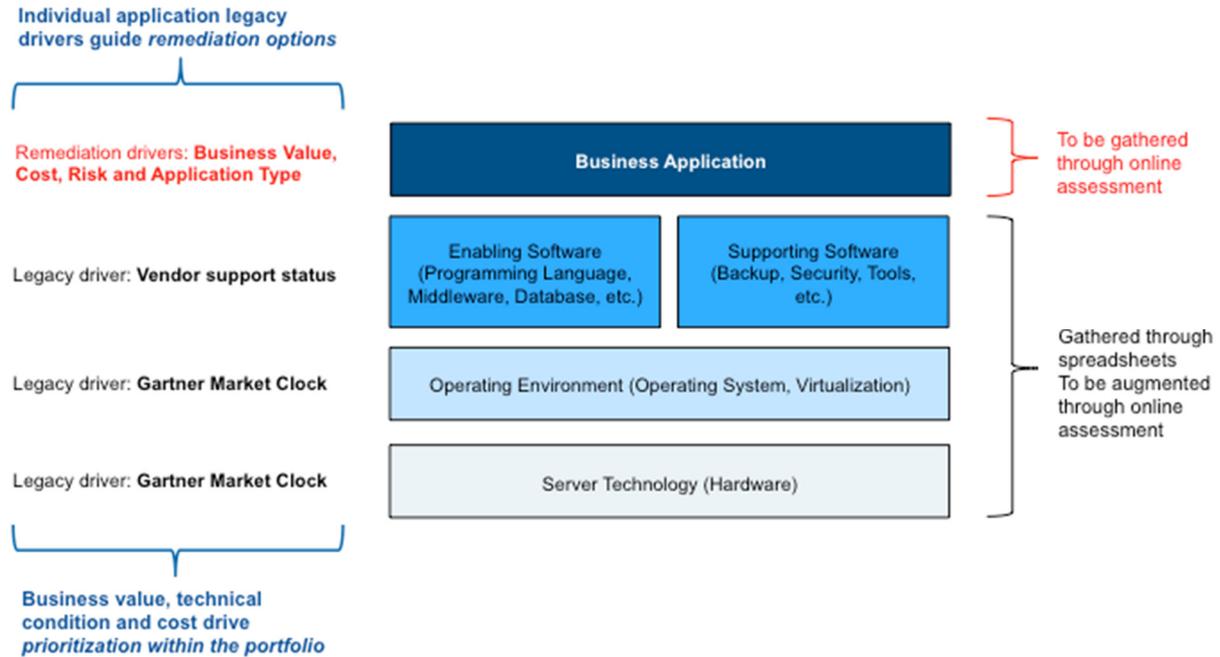


Figure 10 - Augmenting business application context through an online assessment

## Next Steps

The next steps of the LSS analysis work will focus on the following goals:

- Capture the application and technology context
  - Capture the characteristics of the overall business application landscape
    - Driven by business application categorization of all applications (legacy and non-legacy) to be obtained through the online assessment to be filled out by agencies
    - These characteristics will inform the remediation options for consolidation
  - Identify application trends within the broader industry, including product capability consolidation
  - Capture the overall technology landscape characteristics
    - Driven by the software inventory of end-user apps (such as fund management), technology stack components (such as databases) and tools (such as backup and recovery)
    - These characteristics will inform emerging standards and potential to optimize licensing models
  - Identify technology trends within the broader industry, including cloud deployment models
- Perform tools-based analysis of the agency business application input
  - Analysis of business application categorization of legacy and non-legacy applications
  - Analysis of business value, cost, technical effectiveness and architecture of the legacy applications
  - Determining preliminary remediation categorization based on the TIME model
  - Determining the pace layer of the application: whether it is a system of record, system of differentiation or system of innovation
  - Determining the initial prioritization to address legacy applications through application overhaul
- Perform a remediation analysis to determine appropriate remediation options for the high priority targets

The following diagram illustrates these steps within the overall project context:

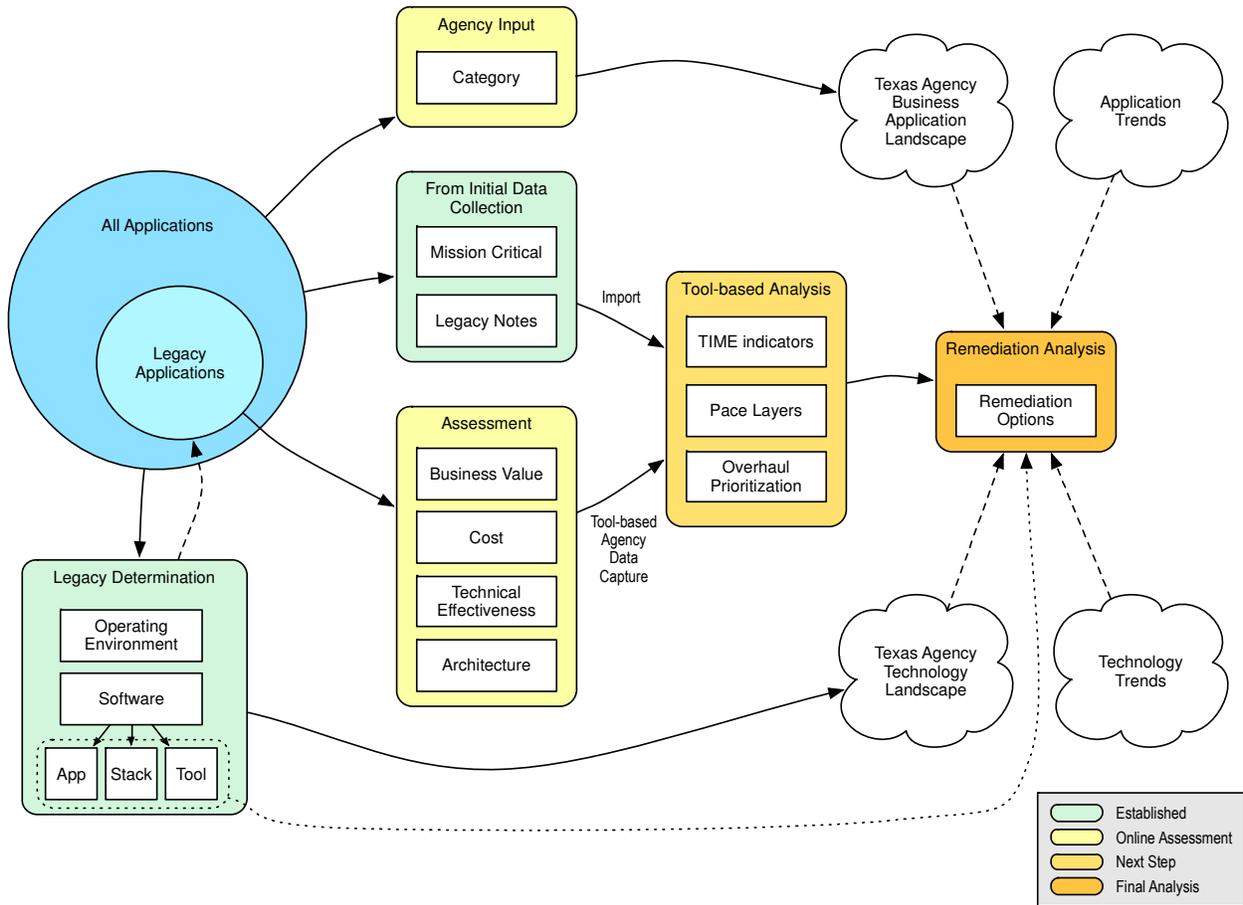


Figure 11 - Next analysis steps

## Appendix A: Analysis Metrics

Between DIR and Gartner, we set a goal of analyzing at least 90% of the normalized software products.

Metric	Value	Description
Unique software entries	10,007	Number of unique combinations of software vendor, product name and version as provided by the agencies and CMDB
Normalized entries	3,361	Number of unique combinations of software vendor, product name and version after the normalization process
Software entries analyzed	3,261	Includes: <ul style="list-style-type: none"> <li>■ Legacy software</li> <li>■ Non-legacy software</li> <li>■ Software that does not affect the legacy status (device drivers, etc.)</li> <li>■ Software with versions that are incorrectly specified or unknown by the vendor</li> <li>■ Software with versions that are not specified</li> </ul>
Software entries not analyzed	100	Includes: <ul style="list-style-type: none"> <li>■ Waiting for vendor response</li> <li>■ Incomplete analysis</li> </ul>
Analysis coverage	97%	Software entries analyzed relative to the normalized entries

Approximately 36% of the normalized software entries have a confirmed legacy status, while approx. 28% of the software entries are current. Note that this number represents the normalized software inventory; it does not reflect the legacy status of business applications.

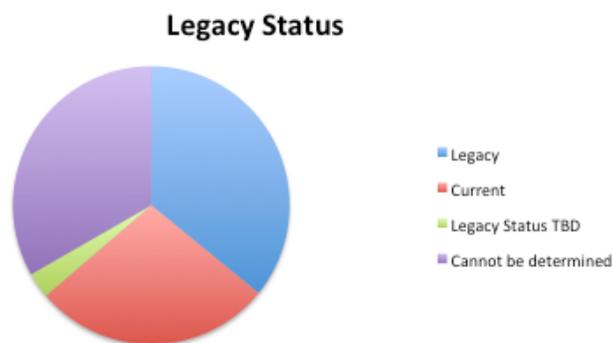


Figure 12 - Normalized software inventory legacy status

## Appendix B: Gartner Market Clocks



Figure 13- Gartner IT Market Clock for Server Technology, 2013

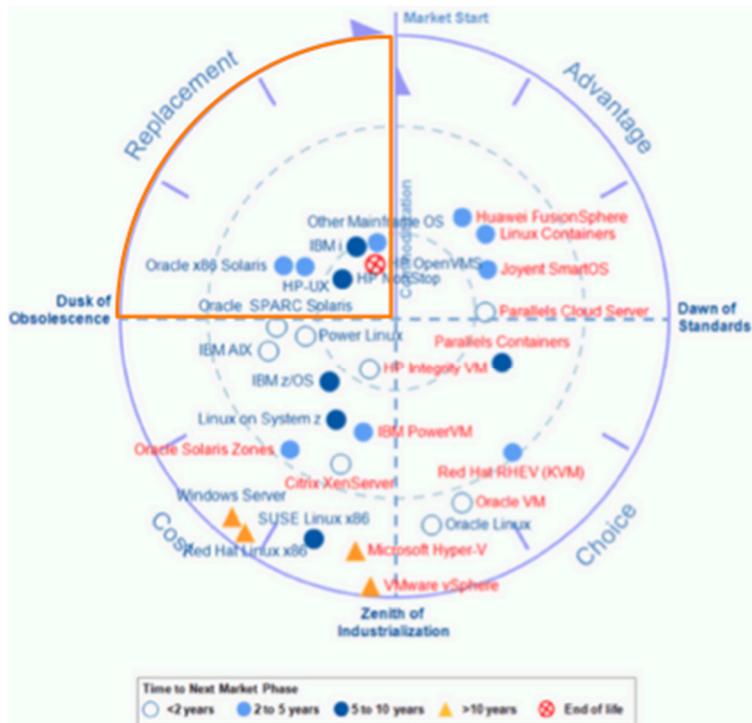


Figure 14 - Gartner IT Market Clock for Server Virtualization and Operating Environments, 2013

**Any questions regarding this Report  
should be addressed to:**

Ernst Rampen  
Director, Gartner, Inc.  
Phone: 310.529.5050  
Email: ernst.rampen@gartner.com

Rob Cohan  
Sr. Managing Partner, Gartner, Inc.  
Phone: +1 512-680-0560  
Email: rob.cohan@gartner.com