

Application Rationalization

What to consider

TIMOTHY MANGAN

TMURGENT TECHNOLOGIES

Introduction

Application Rationalization (**AR**) is a necessary part of any IT strategy and operations, even if it is a nebulous term that is often performed without much of a strategy or plan.

There are commercial vendors with products to address this space, however, it is unclear that they have much penetration in the market let alone how helpful the products actually are.

In this white paper, we attempt to define what application rationalization might entail, in an effort to help your organization build a cohesive application rationalization strategy.

Goals of an AR Strategy

The goals of an organization seeking to build a robust application rationalization strategy may include some or all of the following:

- Software Licensing Cost Management
- Operational cost reduction by removing redundant applications that serve the same purpose
- Operational reduction by identifying troublesome applications, such as those that cause a high ticket count with the help desk.
- Risk reduction by identifying applications or those with a high update count due to software vulnerabilities (CVEs).
- Application Modernization as an aim to improve organization performance.

Software Licensing

The centralized tracking of software license usage is standard in most organizations. This can sometimes be hard as the software to be tracked may not have tracking embedded in it. So Enterprise organizations routinely try to prevent most users from being able to install their own software, depending on software distribution systems that can install the software to account for most of the usage, plus trusted IT Professionals to handle ad-hoc installations with accountability.

Sometimes additional management software can be used to detect application installation and/or usage to aid in further reporting. Typically these organizations “true up” on their licensing consumption on a regular basis.

Additional cost management is possible challenging the end-user organization on the need for the software. This might include justification with the initial request, but it may also include monitoring actual usage of the apps installed as an additional trigger to request justification of unused apps.

Redundant Applications

Every large organization has redundant applications in their application portfolio. Do you really need 5 different ways to read a PSF, open a zip archive, or ftp to some website?

This leads to increased operational costs as each application requires irregular processing of new releases over time. Help desks also might be slightly less efficient as any single technician may not know the nuances of every version of the same kind of application.

Establishing a workflow for initial application requests that considers possible redundancy and challenging the requestor to consider an existing application that might also meet the end-user needs can eliminate some of this from the start.

A periodic review – perhaps once a year- of the entire existing application portfolio might also help catch the possibility of reduction through consolidation as well.

Often there may be a free option and a premium paid option. Consider challenging whether the additional paid features are actually used.

Help Desk Tickets

Some software applications result in a high amount of Help Desk tickets. Your organization probably already has a metric to define a cost for these tickets, and software to help you quantify the impact of each application in help desk (and interrupted end-user) costs.

A periodic review should reveal the most costly applications in the portfolio which could target them for replacement or removal.

Updates/CVEs

There are some applications that update frequently. Which ones are these and why?

This may be because the application is being actively developed/enhanced. Which is generally a good thing, especially with a new product on the market.

This might also be because the application is old and written in technology/languages that are prone to memory corruption and are actively being constantly targeted by malicious actors.

Sometimes it may not be the app per-se, but the framework and libraries that they depend upon that are being targeted and must be updated.

A periodic review of application updates should reveal the top app updates; Look to those apps at the top of the list as candidates for replacement, perhaps by an app not currently in your application portfolio.

Also for consideration in this periodic review is to look at any software with defined end-of-life statements that are approaching.

Application Modernization

It is not unusual for me to be asked to help with an Enterprise migration scenario. The migration might be to a new operating system, to a new style (like Server Based Computing or Virtual Desktop Infrastructure), or a new application packaging and/or delivery system. In those instances we typically start with an application inventory of apps to be migrated. I can tell you that in an organization with more than 20,000 users that inventory will usually be 5 to 20% incomplete. This is usually the most frequent cause of the migration taking longer than planned .

It is a wise idea during such a migration to also consider **modernization** of the applications. While a periodic review of applications for modernization should also be part of an Application Rationalization plan, the migration time when you are already touching all of the apps is an ideal time to include a review for modernization.

- Modernization may include updating to the latest version that a vendor now has. New features may provide new capabilities or improved productivity for the end-user.
- Modernization may also be done by replacing a stale application with a more modern competitor with the feature you need, or even analyzing the end-user workflow to eliminate outdated make-work.
- We can also talk about modernization in terms of how the application is prepared and delivered.

Summary

Application Rationalization is currently a bit of a “black art”, one in which everyone kind-of knows what it entails and do some stuff in this space, but few have a solid strategy and plan for addressing these needs. Hopefully this white paper gives you some things to think about when you formulate or update yours.

PS: Acknowledgement is due to Carolyn Thomas for prompting me to write this white paper.

Relevant references for further research:

- **The Application Rationalization Playbook – CIO.GOV**
<https://www.cio.gov/assets/files/Application-Rationalization-Playbook.pdf>
- **Leanix.net: Application Rationalization Success Kit Free Download**
https://www.leanix.net/en/download/application-rationalization-success-kit?utm_term=application%20rationalization
- **Application Management: What is application rationalization and how to make it a part of your cloud migration strategy | Google Cloud Blog**
<https://cloud.google.com/blog/products/application-development/application-rationalization-what-why-and-how>