

Organizational Considerations When Replacing or Adding a New Software Language

Brian Varney, Experis

ABSTRACT

Replacing or adding a new software language for your organization's users can add a lot of value and productivity to your team(s). There are, however, many factors and considerations to consider avoiding chaos and hidden costs.

INTRODUCTION

This paper intends to outline factors to consider when replacing or adding a new software for your organization's users. Although this could pertain to several different types of software packages, the motivation for this paper is geared towards development/programming software such as SAS®, R, Python, etc.

IT ARCHITECTURE CONSIDERATIONS

To really evaluate the factors surrounding the implementation of a new software package, it is important to define the necessary supporting architecture and topology. There are typically additional costs and talent requirements as complexity is added to an environment.

ARCHITECTURE / TOPOLOGY

There are many important questions to get answered and factors to evaluate when defining the architecture for your new software package. These will involve your own IT department as well as engineers from the software vendor you are considering.

- Will it run in the cloud or on-premises?
- How many servers will it require?
- What server resources will be required?
 - Disk space, CPU, and memory
 - Grid computing
- Where will the data reside that needs to be analyzed/reported on?
- How will the users access the environment?

ADMINISTRATION

Before implementing a new software package, it is important to understand the administrative needs required for everything to run smoothly. Any training or additional talent will need to be evaluated.

VALIDATION

If your organization is regulated by an organization such as the case with pharmaceuticals and financial organizations, it is important to have validation planning and documentation requirements.

WORKFORCE AND DEVELOPMENT CONSIDERATIONS

While the impacts to the IT department are important, so are the impacts on the developers using the software. Evaluating and planning for these impacts can make for a much less stressful roll out.

TRAINING

How will the developers using the software learn a new language while keeping up with their normal work? Which method of training delivery is best for your user community?

- Online self-guided training
- In person instructor led training
- Online instructor led training
- Train the trainer

SUPPORT

The software users and administrators will both need support to have a healthy development environment. Below is a list of things to consider when evaluating support for a software product.

- What is the quality and completeness of the documentation?
- Is there support available and is it centralized or decentralized?
- If a fix or enhancement is needed, is there a process to request it and get it done?
- How much internal talent does your organization have with the software?

PUBLISHING

The results that are needed from a software package need to be considered. For example, if your deliverables require bookmarked PDF documents, you will need to make sure that additional development is not required to achieve those results.

VERSION CONTROL

For results to be reproducible, version control software will need to be considered for the code. Many organizations are turning to Github, as an example. There also need to be snapshot processes in place such that results can be run against data from specified points in time.

CURRENT/LEGACY PROGRAMS AND PROCESSES

If you need to match current results and results from legacy processes, there could be rounding and/or computational differences to code around or explain. If you sunset one of your current software packages, you may have difficulty in rerunning old programs.

AGILITY

How easy is it to obtain and install additional modules for the software. For some software packages, it is a cumbersome and lengthy process. For others, one just needs to download and install.

COST

LICENSING

The licensing costs for software packages are important to evaluate on the platform. Some software packages are free to install and use on the desktop but once they are installed on a server or in the cloud for a larger user base, they require a fee to use effectively.

TOTAL COST AND VALUE OF OWNERSHIP

These are the ultimate measures at the end of the day. What is the total cost of ownership and the total value of ownership. There is also a cost in changing or adding a new development software. The loss of productivity during the learning curve and changing of processes is significant.

CONCLUSIONS

Changing or adding software packages should not be based on purely the licensing costs. There are many other costs and factors to consider and weigh against the added value the new software will bring.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Brian Varney, Solutions Architect
Experis
brian.varney@experis.com
www.experis.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.