The Migration of Microsoft Excel Tools to Next Generation Platforms:

Can You Hear the Footsteps?

Jeremy Eden

Booz | Allen | Hamilton

delivering results that endure

2015 ICEAA Professional Development & Training Workshop

June 9 – 12, 2015

Abstract

Analysis tools are popular solutions for extended deployments and at the same time an acceleration of platform migrations is occurring to keep up with rapidly changing technology and end user demands. This paper provides specific examples of platform migration challenges, resulting failures, and how they can be best addressed. It will also provide specific best practices for ICEAA member take-away that can be followed to help ensure smoother transitions across platforms.

Table of Contents

Executive Summary

Cost estimating and analysis tools are very useful to assist in the evaluation of data, specifically processes involving "what-if" scenarios and decision making. Many tools are created within the Microsoft Excel application of Microsoft Office and allow the implementation of standardized interfaces for analysts and cost estimators without building software applications that frequently require more time, funding, and oversight resources. These analysis tools are helpful and powerful in day to day operations when done well. However, once running, the maintenance of these tools are often ignored or afterthoughts at best to their stakeholders. Such an approach increases the likelihood that one day they will unpredictably no longer function, often when they are needed most.

As the capabilities of modern computing hardware continue to advance at great speed, a shift has occurred within the accompanying software world. Gone are the days that a piece of published software is utilized for several years with only a couple of updates throughout its lifecycle to keep it fresh until the next version of the software is created. Current productivity software trends are for lifespans measured in months rather than years with organizations and users expected to quickly adopt updates over the internet that can come as frequently as every week.

It is up to the developers, sponsors, and users of tools dependent on software platforms to recognize and prepare for these events by:

- Being alert to the timeline and scope of upcoming platform(s) changes
- Adopting plans to accommodate the changes to come
- Implement changes to analytical tools assuring they will continue to function
- If possible, modify analytical tools to take advantage of any new functionalities offered by updated platforms
- Prepare the tool's user community for changes appropriately and completely

New software platforms place a burden on tool stakeholders to assure compatibility with legacy analytical tools and at a minimum guarantee existing functionality is maintained. However, if sufficient preparations are made, the increased capability of the platforms can often be leveraged to extend the usefulness and lifespan of these tools outside of their initial scope.

Introduction

In recent years, analytical tools have become very popular for the repeated and customized analysis of data including cost analysis, risk, logistics, decision making, and strategy. They offer the ability to create customized solutions for users who may not be experts on the processes and practices involved with such analysis, allowing for an increase in productivity and the quality of decision making. These tools often are dependent on a larger software platform or application and leverage built in mathematical functionality to run. Unfortunately, the maintenance of these tools are often afterthoughts for stakeholders, making even the best tools susceptible to failure when the underlying software platforms are upgraded to next generation releases.

Special attention is required by tool stakeholders similar to the approach adopted by software developers in order to helpreduce the impacts to platform changes and in many cases take advantage of them. Additionally, designing tools with planned migration in mind can assist in extending the lifecycle of a tool and making the tool somewhat future-proof. Many times tool developers have unique ideas on interfaces and functionality, but don't consider the stability of a tool should software requirements/user permissions change. Many times restrictions by tool sponsors forbid using resources investigating the ability of a tool to migrate from one software platform to another, even though it has been identified as a risk to the product. A popular saying in the sport of football regarding those who drop the ball often instead of catching is that they "heard the footsteps." It implies that when someone is so consumed with their current challenge, they plan to recognize and adapt to other obvious challenges. It is my hope that this paper will provide specific insight and resources to those who may currently be hearing those "footsteps" and those wanting to avoid such a situation in the future.

This paper will show that it is critical to recognize the need for planning in cost estimating and analysis tools to accommodate software migrations and to improve performance as well as longevity. Presented first will be a checklist of adaptable tool design considerations followed by a discussion of the stakeholder advantages to migrating tools. Next, additional considerations that while aren't critical, should be implemented prior to the migration of platforms will be discussed. Finally, lasting resources will be provided for future reference.

Adaptable Tool Design Considerations

The following tool design considerations and explanations are meant to provide you a checklist of items that can impact the functionality of a tool when it is migrated from one software platform to another. While this list should not be considered a complete compilation of all design aspects, it is a thorough sampling intended to be expanded upon. The actual checklist is a Microsoft Word file, is NOT proprietary, may be used without prior consent, and are available to download from the internet from April 1st, 2015 through June 12th, 2015 using the following address: http://tinyurl.com/EdenICEAAConference2015

Adaptable Tool Design Checklist

- □ **Does the tool utilize Macros?**

  Many tools utilize macros to automate sequences of functions for a user. However, many organizations greatly restrict the usage of macros. Occasionally organizations use application upgrades as a convenient time to implement new security policies for applications and can inhibit the functionality of a tool that was previously permitted to run.

- □ **Does the tool utilize ActiveX Controls?**

  Many tools ActiveX Controls to provide software like controls to users within a tool environment. However, many organizations greatly restrict the usage of macros. Occasionally organizations use application upgrades as a convenient time to implement new security policies for applications and can inhibit the functionality of a tool that was previously permitted to run.

- □ **Has Visual Basic for Applications (VBA) programming been used?**

  Visual Basic for Applications allows the creation of back-end functionality and front end user interfaces that mimic those of more traditional software designs. VBA enabled tools are very powerful, but need to be maintained and monitored in much the same way as traditional software designs. Functionality of VBA controls is known to be expanded, but also has a history of allowing certain functionality to expire.

- □ **Does the tool automatically generate outputs to serve another automated tool?**

  Having another tool dependent on the outputs of a primary tool places an added burden to assure that any outputs will remain constant or if changed, sufficient time, resources, etc. are available for the dependent tool to be modified as well.

- □ **Has the tool been designed to import data from a file or other automated process?**

  If a tool is designed to accommodate the import of data from another file or automated process, it is possible that the newer platform may not accommodate that type of data any longer.

- □ **Is the tool designed to accommodate particular security policies?**

  Security concerns are constantly changing. What was once considered “secure” is often redefined as forbidden in the future. Occasionally organizations use platform migrations as a convenient time to implement new security policies for applications and can inhibit the functionality of a tool that was previously permitted to run.

□ **Has the tool been designed to serve multiple platforms at the same time?**

Tools that are required to run on multiple platforms at the same time can become very difficult to maintain. You must identify what is considered common functionality between the two platforms and what is unique to each platform and plan accordingly.

□ **Is the tool packaged in a utility program for deployment?**

If the tool is packaged in a utility program for deployment to users, it can aid in the deployment of new versions for future platforms. Some packaging utilizes are even advanced enough to check platform versions before installing a tool. However, packaging utilizes can also run into security permission and network policy issues.

□ **Are external references linked in the tool?**

External references linked in tools can automate the inclusion of data into functions without requiring user interaction. However, these links have been known to change in syntax and easily become corrupted leading to failure of tools.

□ **Does the tool interact with other external tools?**

Interaction with other tools is often interrupted by the migration of platforms. Many times tools will use the default functionality provided by the platform it originates on. If interacting with another tool, the receiving tool may not have programming to accommodate the requests from the more advanced tool. Likewise, the more advanced tool may no longer support an older functionality.

□ **Does the tool use data transfer protocols to pass information over networks?**

Passing data over networks allows for great advantages in collaboration. Changes in security policies and protocols for transferring data occur often. A popular network configuration and collaborative environment today may not be the one utilized tomorrow.

□ **Are charts and other graphics generated by the tool?**

New graphics are always becoming available and in demand. Sometimes older graphics are abandoned and no longer offered by a platform and need to be checked often.

□ **Do all users of the tool migrate platforms at the same time?**

Many organizations, especially larger ones will conduct the upgrade of platforms in a phased approach with two platforms supported for a certain amount of time. Question if the tool will need to support one platform or both and when the change will be labeled complete and older platforms no longer have to be supported.

☐ **Is customized user input data saved and reused from one tool session to another?**

If user input data is being stored, all future versions of a tool must be built to accommodate the import of that old data. If the old data what created using an older platform, it may be challenging to import it to the newer tool on the newer platform.

Stakeholder Advantages to Migrating Microsoft Excel Tools

The migration of Microsoft Excel tools from one platform to another provides many advantages to all tool stakeholders including the developer of the tool, the sponsor of the tool, and to the end user. The following provides a sampling of these advantages from the viewpoint of each stakeholder.

Tool Developers

Constraints placed on tool developers often leave little room for the contemplation of maintenance and support in the future, especially for platforms not yet released. After all, the one advantage of these tools is that they do not require the same rigor as fully developed software. Developers are now expected to provide tools that will continue to function long after delivery and require only slight modification to accommodate new operating conditions or requirements. Often, it is desired that the tool be designed in such a way that a non-developer administrator can organically update the tool with little or no training in the future at a minimum of expense and time. The design of tools with specific attention paid to future migration empowers the tool developer with many advantages including the following:

- The ability to plan ahead and schedule expansion and modification of tools rather than using “quick fixes”
- Understanding that documentation of tools is not just historical, must be as complete as possible, and may be used for reference in the future to implement a change
- Incentive to add developer features to the tool that may not be used often, but can assist in future migration and be worthwhile to the maintenance of the tool
- Insight to avoid certain development practices that can be advantageous for tools not requiring migration in the future, but cause unwanted constraints and obstacles in the future for tools that will undergo a platform migrated
- Ability to provide a tool to the user community that limits interruption and takes advantage of new platform functionality as it becomes available in new platforms

Tool Sponsors

Tool sponsors, especially those in the United States Government, are often at the mercy of their own Information Technology (IT) department when it comes to the software platforms installed on workstations. By enabling tool designs to accommodate future software platforms without extensive modification, the tool sponsor is now empowered by new platforms rather than challenged by it. Adoption of future software platforms has many advantages, but can also lead to difficulty as the timeline for the change in platforms varies greatly from organization to organization. Adopting new platform capabilities too soon can create as many problems as adopting them too late. Because of these challenges, stakeholder often fear software platform changes. However, if correctly anticipated, the migration of a Microsoft Excel tool to a next generation platform will provide the following advantages and become something encouraged rather than endured:

- The ability to continue receiving a return on the initial investment of developing and maintaining a tool
- Assuring minimal or no interruption in process or tool performance due to platform changes
- Showcasing the Microsoft Excel tool as a professional product with design and deployment capable of advancing with technology
- Enabling the sponsor to consider future tool expansion and enhanced capability rather than "patching the bugs" resulting from migrating the tool
- Continued return on investment of tool support products tool such as training classes, curriculum, metadata, documentation, and websites

Tool Users

Tool users represent the stakeholders most dependent on timely tool migration. The day-to-day user endures "fire drills" for analysis by using tools to generate efficient and accurate analytical results without interruption. Should the requirements of a tool not be completely in compatible with the platform used by a particular user workstation, a tool failure can result leaving the user with no product available until the platform is upgraded to match the tool or vice versa. As multitasking and analysis is pushed to limits, the advantages of appropriately timed tool migration becomes a *critical* for the end user to provide the following:

- Confidence that work performed using a tool today can be replicated and completed tomorrow
- Ability to continue using the tool without significant interruption due to platform changes
- Satisfaction that the developers and sponsors of the tool are continuing to support the tool and its users
- Continued return on training, skills, and time invested to use the tool

Additional Considerations

The prior section of this paper introduced many points for the migration of Microsoft Excel tools. However, best practice demands that other considerations be taken into account when assessing tool migration impacts. Below are additional factors that should be also be explored prior to migration to new platforms.

Documentation

Although most tools involved in this discussion are not formal software applications, properly developed tools do include extensive documentation. Requirements, user manuals, tool tips, systems manuals, training curriculum, and study reports are all common sources of information that should be considered perishable in the event of tool migration. Proper scheduling for updates of these documents must be incorporated into migration plans so that support for the tool can continue with minimal disruption and all stakeholders do not find documentation that contradicts the deployed tool.

Security and Data

Security of tools should always be a high priority both in the initial development and in sustained maintenance. Just because a tool was secure when it was first developed does not mean it will remain that way forever. Likewise, just because the data a tool utilizes was originally considered appropriate for distribution within a tool does not mean that it will forever remain that way. An example of such a scenario is that at one point in time Social Security Numbers were considered safe for distribution. This of course this is no longer the case today.

Testing

Testing should be conducted for all functions and operations within a tool when it is migrated to a new platform. This can be quite tedious as it requires an extensive time to check every possible selection/input for every field and function of a tool. However, you cannot rely on an obvious error of function or operation to indicate what in the tool must be altered. There may be hidden errors and as well as errors that will not occur until other functions are upgraded to the new platform. It can be increasingly frustrating to all stakeholders to be assured that a tool is ready for use on a new platform and then find it is not. In an even worse situation, developers can find themselves caught in a loop of patching individual errors in a tool, assuming they have addressed all issues for the new platform, and once released find that more work must be done and another new release generated.

Packaging

Packaging of tools in standard Microsoft Windows Installer Packages (MSI files) is an excellent practice that can return great benefits to stakeholders. These packages allow the deployment of tools in an organized fashion that supports configuration management, version upgrades, automated removal of old tools, icon/shortcut integration, and placement of multiple tool files. Such packaging also gives a tool a professional look and supports the user interface experience for the end user from the moment a tool is obtained. Tool packages also allow the identification of which platforms stakeholders may be utilizing to assure that tools are only used on those platforms supported at the time.

Conclusions

As discussed in the Adaptable Tool Design Considerations section, Microsoft Excel tool migration considerations provided in this paper are only a small sampling of possibly dozens of things to consider for overall tool migration plans. However, the important thing is *to begin* considering platform impacts on tool migration. Many find the process overwhelming and unnecessarily formal for these types of tools. However, to simply ignore the issue is inviting failure and frustration that might have been avoided with surprisingly little effort. Like so many processes, approaching the obvious concerns first and then digging "deeper" into the details as you move forward will yield vital experience that cuts down on time and effort for such assessments in the future.

Many tool developers, sponsors, and users are frustrated by the resources required for maintenance and many times do not conduct this maintenance for tools that have been designed to avoid the normal burdens of developed software. If one can insert planning for the migration of tools to newer platforms from day one, the resources and effort required when the time comes can often be minimized. The most often reported obstacle is not making adjustments to such tools, but identifying what adjustments need to be made to comply with a new platform. To address these concerns, I highly recommend the resources listed in Appendix I: Tool Resources for Migrations to New Platforms below. Most resources are in internet message forums populated with experts willing to help. When new platforms are distributed, many tool stakeholders find themselves in the same predicament of migrating their tools so common issues are identified and solved quickly. My hope is that the concepts mentioned in this paper and listed in the accompanying considerations will convince tool developers and users to avoid *Hearing the Footsteps* and run with the ball rather than drop it. I have further expectations that this discussion will help generate meaningful discussion between tool developers, sponsors, and tool users so resources are put into the development and maintenance more readily and effectively leading to more adaptive tools.

Appendix I: Tool Resources for Migrations to New Platforms

Programming Resources

Microsoft Office Compatibility Checker
https://support.office.com/en-us/article/Check-a-workbook-for-compatibility-with-earlier-versions-of-Excel-cc9d3b72-8b2a-428f-a81f-ce656837fcf3

Microsoft Answers Community
http://answers.microsoft.com/en-us/office

Support for Microsoft Office Community Forums
http://support.microsoft.com/gp/gp_newsgroups_master

Microsoft Developer Network Visual Basic for Applications Forums
http://social.msdn.microsoft.com/Forums/en-US/isvvba/threads

Mr. Excel.com
http://www.mrexcel.com/forum/forum.php

Excel Forum
http://www.excelforum.com/

The Code Cage Forums
http://www.thecodecage.com/forumz/index.php

Security Resources

PC Tools by Symantec Secure Password Generator
http://www.pctools.com/guides/password/

Installer Package Resources

Caphyon Advanced Installer
http://www.advancedinstaller.com/

Other Resources

Full Adaptable Tool Design Checklist (Available April 1st, 2015 through June 12th, 2015)
http://tinyurl.com/EdenICEAAConference2015

Of course… Google
www.google.com