Streamlining application compatibility testing at Microsoft

The way Microsoft delivers products and technology has evolved. Rather than releasing a single large product update every few years, Microsoft has moved to a more frequent release cadence with smaller updates. Before Microsoft IT can deploy any new products or technologies within our enterprise environment, we must first determine compatibility with our critical line-of-business (LOB) applications. The success of many mission-critical business processes relies on our ability to maintain a stable, predictable, and consistent application experience.

As product release cycles evolved, so did our testing methodology. We focused on streamlining our testing model to support the accelerating technology development and release cycles at Microsoft. We’ve created and refined our application compatibility testing program to increase our testing efficacy, while reducing the time, cost, and effort of each test cycle. To more rapidly adopt products and updates, we optimized our processes, rationalized our application portfolio, identified critical applications and golden scenarios for targeted testing, continued to analyze failure trends over test cycles, and leveraged virtualization and automation. What used to take us months to complete now takes us weeks—and with fewer resources. Lastly, we’re developing an application modernization strategy to help our application portfolio leverage the latest capabilities and features that offer improved security and better performance and reduce the need for compatibility testing.

Shifting how we think about testing

Traditionally, application testing methodologies included testing everything that could be potentially impacted by the deployment of any new product, update, or platform. In an enterprise, finding out who can perform all of that testing is not an easy question to answer when you are looking at testing everything. We began to see the need for a more agile approach—testing less but still having a predictable outcome for deploying new platforms. With the new services model, this approach is more important than ever.

At Microsoft, nearly all segments of our business have applications that are considered critical. Those applications are typically specific to that segment of the business, such as Finance, Accounting, Human Resources, or Manufacturing. The shift in our testing strategy was motivated by our need to find issues as early as possible to manage risk to our critical applications and to mitigate potential losses of productivity.

We have seen a consistently high pass rate for application compatibility since Windows 8, averaging 98 percent in Windows and Office 365, 100 percent in Internet Explorer, and 100 percent in Microsoft Edge for modern browser compatible applications. This pass rate has continued through Windows 8.1, multiple Windows 10 updates and several Office updates in the recent years. We attribute the continued pass rate success to several contributing factors:

- Windows versions and upgrades have maintained a high level of backward compatibility for Window 7 and Windows 8.
- Internet Explorer 11 has been compatible and is available for fallback for legacy web applications that have not been modernized.
- Office 365 has smaller, incremental updates that are released monthly. Reducing the amount of change with each release makes it easier to identify potential issues.
Driving down costs and increasing application compatibility testing efficiency

Each time we deployed a new product or release, such as an operating system, productivity tools, or an Internet browser, the application test teams would have to interrupt their internal maintenance and release schedules to perform the necessary testing to ensure those applications were compatible with the new technology. Depending on the number of LOB applications in an application portfolio, many teams had to hire additional resources to keep up with necessary application compatibility testing. To introduce efficiencies and make the best use of resources, we:

- Created a centralized team of dedicated testers to test the critical primary applications.
- Worked with application teams to identify a number of critical primary applications within the LOB application portfolio for targeted testing.
- Developed an application portfolio management tool to track detailed information about all the applications in the LOB application portfolio.
- Moved bug tracking to a centralized system.
- Combined individual test cases into golden scenarios, combined platform testing, and leveraged automated where available.

Centralized team reduces resource requirements

Dedicated testers are more effective in their role, and can look at issues and trends with an objective eye. Because testing is their primary function, they have the agility to keep up with the increased cadence of Windows and Office releases as both platforms move from products to services. Our dedicated test teams can meet the demands of the business and minimize impact to the organization.

To facilitate better communication, the centralized test team acts a single “quality partner” who works with application owners and their distributed application test teams.
We also track test results from each test pass and share with all application owners. One of the benefits is that we can analyze pass and fail trends across applications—and over multiple releases—that we make available across application owners. The application teams have distributed test teams that use our results to test their critical secondary applications. The distributed test teams are the experts on individual or groups of applications and can test them to a deeper level. Their test pass can usually be completed in about three weeks. Historical trends, including most recent failures, help us better plan our testing needs and provide visibility into the potential for future failures.

Trending data shows that there are year-over-year reductions in the cost of testing each application. A single tester can support many applications during each test pass, saving time and reducing cost significantly over a more distributed testing model.

App rationalization and targeted testing

With monthly releases, we simply don’t have the resources to test everything. To minimize the impact on application teams and mitigate the risk of application failures that can increase support costs, we developed a predictable plan for testing and optimize our testing efforts through a sample-based, targeted testing program.

We worked with the application teams across the organization to create a list of business-critical primary applications for the centralized test team to focus on during their test pass.

There are approximately 2,100 applications in our LOB portfolio. To increase our efficiency, we identified approximately 500 critical applications:

- 300+ applications are critical primary applications to be tested by the centralized test team.
- 200+ applications are considered critical secondary applications that are tested by the distributed application test teams.

To determine which applications should be included as a critical primary application, we prioritized them based on business criticality, categorized them based on technology, and chose applications that represent others. These applications are cross-referenced by technology dependency, and data from previous testing is used to identify applications with past issues.

The critical primary applications have commonalities with other applications, so they serve to represent several other applications in the portfolio. When a critical primary application passes, we can hypothesize that similar applications will also pass when tested.

Managing third-party applications

We have a number of third-party applications in our LOB application portfolio. We do a full rationalization of our inventory of third-party applications using our software licensing and software asset management (SAM) processes. Our SAM processes cover the entire lifecycle of third-party software that we evaluate, purchase, deploy, use, manage, and retire. For critical third-party applications, we typically have service-level agreements that products will be compatible with our standard configurations. We make sure we’re running the latest release of third-party applications and work with suppliers when there are issues.

Some of the suppliers we work with are involved in early adoption efforts, and have access to pre-release builds that they can use to test their applications. They can determine whether their applications are compatible or if they need to apply a fix or update. Early adopters can provide valuable, and often actionable, feedback to the product groups.

Refining by platform

We test against our primary platforms, Windows, browser, and Office. We select approximately one-third of our critical applications to be tested for each platform. Approximately 10 percent of our total LOB applications get tested on each platform in an average test pass. Some applications are tested on more than one platform.
Table 1. Example primary and secondary applications mapped to platforms for testing

<table>
<thead>
<tr>
<th>Application name</th>
<th>Platforms application must be tested on</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Windows</td>
</tr>
<tr>
<td>Application 1</td>
<td>Yes</td>
</tr>
<tr>
<td>Application 2</td>
<td>No</td>
</tr>
<tr>
<td>Application 3</td>
<td>Yes</td>
</tr>
<tr>
<td>Application 4</td>
<td>No</td>
</tr>
</tbody>
</table>

Tracking information in the application portfolio management tool

When a new LOB application is installed, created, or deployed at Microsoft, including third-party applications, it’s added to the application portfolio management (APM) tool. Built using Azure SQL Database and SharePoint Online, we use the APM to track information about all the LOB applications that are in the environment, including business criticality, number of users, and key testing data. By using the APM to track data points and trends, we can identify and predict certain types of errors or bugs in an application, and bugs that might occur across similar applications. The systematic collection of LOB data gives us a clear picture of applications with frequent issues, as well as applications that pass consistently. By tracking and applying this data, our test team can analyze trends and generate reports with reliable predictions of future compatibility issues.

Centralized bug tracking improves visibility and collaboration

Each application team may have their own bug tracking system. However, to better understand the bigger picture of where and when bugs occur, we track our own bugs using Visual Studio Team Services (VSTS). Using these bugs gives us visibility into product bugs and applications that routinely have issues, so we can work with application owners to investigate and eliminate bugs across applications that have the same resolution.

Additionally, data trends can help guide testers to understand what’s changed when a new bug occurs that hadn’t shown up in previous tests. This also allows them to track where bugs occur in similar applications that might share code. We can view whether there were failures with a specific application by viewing historical bug tracking data. This information helps us prepare for potential issues in future test passes.

Using a dashboard for self-service reporting and analytics

We use a Power BI dashboard that pulls application information from the APM tool and bug information from VSTS to give us a holistic view of the application compatibility program. The dashboard overview is a useful view that we share with leadership and decision makers to give them a view of testing schedules and the compatibility results against the various platforms.

Application owners and product teams can use the dashboard as an entry point to access all the self-service reporting and analytics they require. This has been particularly useful in reducing the number of random requests for views of that information that we used to receive prior to the availability of the dashboard.
From the main dashboard, we can drill down to product specific dashboards that focus on more granular details, like compatible count, test pass progress, participation details, and issue analysis. In this view, we can review the highlights or summary of the test passes and other activities we’ve done recently. The ability to view both overview and detailed information increases the speed in which we can detect and respond to any trends or issues that arise.

Automating testing

We receive the full suite of test cases from the application teams, and combine and modify them to create “golden scenarios.” To make the best use of our time and resources, we use the “golden scenarios” to combine the testing of the key capabilities for which the application was created, rather than testing only a single capability at time, and avoid redundant test cases. We only use about 20 percent of the test cases per app; for a major deployment, we can complete the test pass on all the critical primary applications in a week.

We automate test cases whenever possible to further improve testing efficiency. Automation provides for much quicker testing, resulting in more applications tested in the same period. We have full automation for approximately 50 percent of our test cases with the rest of the applications requiring manual testing.

We only rely on manual testing for things that can’t be automated; for example, we use it with applications that experience frequent changes to the user interface. Using VSTS, we can record cursor clicks and data entry during manual testing and to “replay” those steps in a script for subsequent testing cycles to automate the steps they followed in testing. The scripts serve as a partial form of automation and have helped us minimize the overall level of effort involved in manual testing.

The output of our automated and scripted manual testing includes reporting that contains all the information required to create a bug in the centralized bug tracking system. Because of the efficiencies we’ve gained through automation and scripted manual testing, we’ve expanded application coverage by 400 percent.

Virtualization provides more consistent and scalable test environment

To increase efficiency, we use an Azure subscription-based virtual machine testing environment with the Azure DevTest Labs service. Using the service, testers perform a large number of tests quickly without setting up or managing a physical environment. We can easily change the scope and scale of our testing activities, and change platforms or configurations without impacting any physical resources.
The consistency and predictability of the virtual machine environment also limits the introduction of variables that can cause false positives and slow down testing times. It’s very easy for us to reproduce test environments. By using a virtual machine test environment, we can also quickly expand or reduce testing capacity as necessary with minimal investment of time or resources. This is an advantage when we are addressing last-minute requests or schedule changes.

**Application modernization**

Managing legacy applications is resource intensive and the costs of ongoing maintenance are consistently increasing. To support our increasingly mobile workforce, we’re also in the process of migrating many of our LOB applications to the cloud. Not all applications can be successfully migrated as-is. We encourage application owners to take that as an opportunity to consider modernizing the application so that it can connect and interoperate with newer platforms and ecosystems.

From an application compatibility testing standpoint, modern applications can be tested more efficiently, yield fewer bugs, and offer easier issue remediation. In simplest terms, when you modernize your applications, most compatibility issues go away.

In an enterprise, where the total LOB application portfolio can include thousands of applications, it can represent a significant amount of effort, time, and expense to try to modernize all the applications that are in use. We have a program in place with our application teams to help them assess the current modernization status of their application portfolios and help them decide where to focus their efforts in modernizing their applications.

Most LOB applications at Microsoft fall into three categories:

- **Key applications.** As identified by the applications owners, key applications are either already modernized, or most have modernization plans in place.
- **Sustain applications.** Whether there are budget constraints or there are future plans to upgrade or replace, sustain applications are the legacy applications still in use, with no immediate plans to be modernized.
- **Sunset applications.** Sunset applications are still in limited use, and have a roadmap to retirement.

For desktop applications, our approach is to design and leverage built-in controls and capabilities of Windows, and have improved interoperability with Azure cloud technologies. For mobile experiences, our strategy includes cloud-based cross platform mobile applications that are accessible and perform at scale. We’ve pulled together standards, guidance, **design principles**, and best practices for developing modern applications including the **Universal Windows Platform**, that provides a common app platform for modern applications with a consistent experience across all Windows devices.

Many of the LOB applications at Microsoft are browser-based, so we do a lot of work with application teams to help them determine where they should look to take advantage of the capabilities of the new Microsoft modern browser, Microsoft Edge.

Over the past 20 years, enterprise organizations and independent software vendors have standardized and developed their web-based applications and services on specific versions of Internet Explorer. Some of those applications and services are dependent on Internet Explorer and the proprietary technologies that it supports, such as ActiveX Controls and browser helper objects.

Although legacy applications will continue to work on Internet Explorer 11, we are encouraging the modernization of critical and key applications. Microsoft Edge offers faster browsing, better security, and new features like Reading View and Cortana integration.

**Conclusion**

We’ve evolved our application compatibility testing into a streamlined process that’s efficient and agile. We can perform more testing with fewer resources, more quickly. We can stay on pace with monthly Office 365 updates, and quarterly Windows upgrades, because our test passes take weeks, rather than months to complete.

[microsoft.com/itshowcase](microsoft.com/itshowcase)  
December 2016
Centralized test teams can meet rapid-release cycles without impacting ongoing development and project-related work of the application teams. Dashboard-based self-service analytics and reporting has reduced the amount of time and effort we used to spend answering service and status request from business groups and application teams.

Centralized bug tracking gives us current and historical bug tracking, so we can easily see status of logged bugs or anticipate potential issues based on prior test results. The centralized bug tracking also increases visibility between the product group and the applications team, making bugs more actionable, earlier.

Looking ahead to the near future, we’re looking forward to using telemetry and Upgrade Analytics to capture more details about the computing environment. Telemetry will surface trends that we can’t yet see to help us better recognize pass/fail patterns and prioritize what to test. We’ll have more visibility of application presences and usage, that will help us rationalize our application prioritization based on real data.

For more information

Microsoft IT

microsoft.com/itshowcase

Taxonomies for Categorizing Apps to Drive Your App Compat Project

© 2016 Microsoft Corporation. All rights reserved. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners. This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS SUMMARY.