Application Compatibility Framework - Building Software Synergy

Shishira Rao
Amrita Desai
Ashish Khandelwal

~McAfee Inc.
Agendum

• Setting the Context - What, Why and Importance of Compatibility Testing?
• Application Compatibility framework
• Three testing types (OS, 3rd Party & Endpoint)
• Implementing and sustaining the Framework R-over-R
• Challenges & Future Action Plan
Compatibility testing is the process to determine the impact of conflict between multiple objects and to maintain information system functionality as intended.

“Every Product is subject to Risk………..”

- Downgrades to XP due to Vista incompatibility issues
- Suzuki Alto aka Maruti A-star recalled in Europe for Tail light problem
- Website render wrong in some browsers
- 11.8 million Dell Optiplex computers might break or cause fires due to faulty capacitors
A Broader Categorization

1. **OS Compatibility**
   - Windows 7
   - Windows Vista
   - Linux
   - Apple

2. **3rd Party Compatibility**
   - Microsoft Office
   - Skype
   - Firefox

3. **Endpoint Compatibility**
   - McAfee
   - Norton
   - Bitdefender
# Application Compatibility Framework

## Build Released to QA
- Gathering Product Knowledge
- Understanding User Story

## Test Suite Preparation
- Test Plan and Test Scenarios Creation

## OS Compatibility Testing
- Identify Compatibility Checkpoints
- Running OS Compatibility testing tools

## Endpoint Compatibility Testing
- Endpoint DI Identification
- Pilot Program (PP)

## Third Party Compatibility Testing
- Identification & Categorization
- Prioritization & Execution

## Reporting & Analyzing Results
- Test execution Metrics
- Defect Metrics
- Result Analysis

**DI** - Defect Identifiers  
**OS** - Operating System
“OS behavior change can propagate numerous compatibility defects in an application. The purpose of this testing type is to uncover issues which occur due to these changes.

Question 1: Will your Regression not support a new OS Support Case?

Question 2: If the answer to Question 1 is Yes, where does OS Compatibility come in here?

Most OS support claim from QA goes with the intention of confirming the Product features fit well within the OS framework.

Do you also plan for what OS features (you are ignorant of) could boil up your defect database?
OS Compatibility testing Life Cycle

**Identify**
- Capture the OS support info from your MRD
- Identification of Compatibility Checkpoints on a OS
- Identification of tools such as ACT, Verifier.exe, IE Compatibility test tool

**Research**
- Study Compatibility checkpoints to do feature risk analysis
- Identify features affected by checkpoints
- Study tools and identify scenarios

**Perform**
- Create test scenarios with info collected from the previous stage
- Execution of test scenarios (When should you do this?)
- Run the identified scenarios using compatibility tools

**Report**
- Report and log issues found
- Analyze results and archive test cases for regressing upcoming builds
- Documenting Compatibility Checkpoint table

**OS** - Operating System
**ACT** - Application Compatibility Test Tool
**MRD** - Market requirements document
Compatibility Checkpoints

• What it is?
  OS features which can be point of conflict with your product modules.

• How to identify?

Analyzing

a) OS release document for new features
b) OS known application compatibility issues
c) ACT Tool Community/Vendor Assessment (only applicable to Windows OS)
OS Compatibility tools

“Use to find defects using compatibility tools available in the market provided by the OS vendor.”

Few e.g. are below

Verifier.exe
Driver compatibility check tool with checks like Deadlock Detection, IRP Logging, I/O Verification.

Application Verifier
Application compatibility tool with checks like Memory usage, low resource simulation, deprecated APIs handle.

IE Compatibility test tool
IE compatibility test tool is a part of ACT tool which identifies the compatibility issues of websites with different browsers.
Win Vista – OS model approach

Identify

- OS: Win Vista
- Checkpoint: User Account Control

Research

- Administrative privilege Operations
- Writing to global Locations
- Custom Installers, uninstallers and Updaters
- UAC Virtualization

Perform

- Install/Uninstall of product with UAC ON
- Logging/Events generation
- Monitoring system process

Report

- Documenting issues
- Compatibility checkpoint table
- Maintaining test suite for regression

Checkpoints example for vista

OS versioning
Windows Resource protection
IE Protected mode
Session 0 Isolation
IPv6
“This testing type ensures our product compatibility with those applications that are written by other companies and do not come by default with OS.”

Issues

- How to identify which 3rd party Applications to involve in your testing?
- Does your inventory include customer applications?
- Can applications having same underlying technology be combined and share a common test suite?
- What guidelines to perform compatibility testing of these applications?
- What is the criteria to chose mission critical applications?
Application Identification

- Identification of Products on the basis of various characteristics

Application Categorization

- Categorization of Products based on Category and relevance

Application Prioritization

- Prioritization of Products based on Popularity and usage

Execution & Reporting

- Execution, defect reporting & Metrics Graph collection

**Application Identification**

- a) Application Compatibility Toolkit (ACT) to identify applications installed on a specific system.
- b) Tool/Ways to identify application installed in customer environment
- c) Internally reviewed and installed product repositories
- d) Any Enterprise Inventory tools
Application Categorization

Procedure to group applications based upon their underlying technology. e.g. WinZip, WinRar belong to Archivers category

**How to do?**

Identify categories relevant to your product.
Define guidelines per category to create a test suite.
Group your selected application in one of the category.

**Sample Category list**

<table>
<thead>
<tr>
<th>Application</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antispyware</td>
<td>Data Encryption</td>
</tr>
<tr>
<td>Antivirus</td>
<td>Email</td>
</tr>
<tr>
<td>Archivers</td>
<td>Data Recovery</td>
</tr>
<tr>
<td>Browsers</td>
<td>Firewalls</td>
</tr>
<tr>
<td>CD-DVD Tools</td>
<td>Folder Tools</td>
</tr>
<tr>
<td></td>
<td>Instant Messengers</td>
</tr>
<tr>
<td></td>
<td>Multimedia</td>
</tr>
<tr>
<td></td>
<td>Networking Tools</td>
</tr>
<tr>
<td></td>
<td>Office AddIns</td>
</tr>
<tr>
<td></td>
<td>TDI Applications</td>
</tr>
<tr>
<td></td>
<td>Toolbars</td>
</tr>
<tr>
<td></td>
<td>VPN Applications</td>
</tr>
<tr>
<td></td>
<td>Printers and Faxes</td>
</tr>
<tr>
<td></td>
<td>System Tools</td>
</tr>
<tr>
<td></td>
<td>Communication tools</td>
</tr>
</tbody>
</table>
Application Prioritization

- Prioritize applications based upon various factors in order to streamline efforts to cover most critical applications early in testing cycle.
- **Demo (Ruby Script for Prioritization)**

### Popularity & Usage
- Application popularity by Hits in Search engines (Google, Bing)

### Defects analysis
- Defects logged in previous Product releases
- Tracking of issues with “Compatibility” keyword in Bugzilla

### Customer escalation
- Product DL mails
- Beta Forums
- Defect tracking tool
Application Prioritization (cont)

Popularity Hits (PH)
Firefox has 801m, Chrome has 622m and IE has 1.82b hits. So Priority Links of Firefox can be calculated as:

\[
\text{#Hits of Firefox/Total #Hits} = \frac{801}{801+622+1820} = 0.246
\]

Defect Fraction (DF)

# of Defects of an Application/Total # of Defects of All Applications under that category
E.g. Firefox has 4, Chrome has 3, and IE has 3 defects

Defect Fraction of Firefox = \[
\frac{4}{4+3+3} = 0.4
\]

Customer Escalation Fraction (CF)

# Of Escalations of an Application from Customer / Total # of Escalations of All Applications under that category

For E.g., Firefox has 1, Chrome has 0 and IE has 1 escalation.

CF of Firefox = \[
\frac{1}{1+1} = 0.5
\]

Priority Points (PP)
Firefox Priority Points = 0.246 + 0.4 + 0.5 = 1.146

Similarly, IE & Chrome Priority Points can be calculated as 1.361 and 0.492 respectively.
**Case Study (3rd party application testing cycle)**

**Phases**

1. **IDENTIFY**
   - Gather application inventory using the Inventory tools (ACT) & Customer environment analysis.
   - Research on feasibility of testing
   - Gather Information of applications & it’s features.

2. **CATEGORIZE**
   - Grouping of Products on the basis of underlying technology
   - Identify integration points
   - Create of guidelines
   - Prepare Compatibility test plan and scope

3. **PRIORITIZE**
   - Prioritize applications on the basis “Priority points”.
   - Map the guidelines with the Prioritized applications
   - Select prioritized applications for testing on the basis of scoping.

4. **EXECUTE**
   - Execute test suite on high priority applications in all categories.
   - Map actual results with expected and report issue in case of any variation.
   - Follow up on reported issue

**Activities**

**Deliverables**

“List of applications for Compatibility is finalized”
1. Mozilla Firefox
2. Internet Explorer
3. Google Chrome
4. Norton Antivirus
5. Trend Micro Antivirus
6. McAfee Antivirus(VSE)

“Grouping of applications is decided”
**Category: Browsers**
1. Mozilla Firefox
2. Internet Explorer
3. Google Chrome
**Category: Antivirus**
1. Norton Antivirus
2. Trend Micro Antivirus
3. McAfee Antivirus

“Applications are prioritized & final list is ready for testing”

**Priority1(P1) Applications:**
1. Internet Explorer
2. McAfee Antivirus

**Priority2(P2) Applications:**
1. Mozilla Firefox
2. Norton Antivirus

“Execution of applications according to Guidelines and reporting of variations”

**Browsers:**
1. Verify add-in integration in IE & Firefox
2. Verify basic functionalities of Browser and AUT.

**Antivirus:**
1. Verify AUT dll integration in anti viruses.
“Endpoint compatibility testing is a process that enables synchronization of point products to run on a computing environment.”

Our endpoint compatibility model avoids an expensive pitfall and ensures that a conflict does not exist in the customer's environment with the set of endpoint products already installed that renders the product inoperable.

**Identify**
- Endpoint products & versions
- OS execution plan

**Design**
- Defect Identifiers
- Test suite creation & review

**Perform**
- Create test environment
- Test execution

**Reporting**
- Test Results Tracking & Defects reporting
- RAG status

OS - Operating System
RAG - Red Amber Green

17 Application Compatibility Framework - Building Software Synergy
Confidential McAfee Internal Use Only
Attributes

1. Inter-Team collaboration
   a) Knowledge Acquisition
   b) Feature Flash documents

2. Defect identifiers
   a) Identify collision & conflicting areas
   b) Test suite creation

3. Pilot Program
   a) Customer environment simulation
   b) Large scale deployment
   c) Concurrent endpoint testing
Case Study – Endpoints Products in sync

Identify
- Released version (RTW) of other Endpoint products
- Beta version of Product under test
- Operating System information

Design
- Product specific Processes and Registry entries
- File / Folder Protection modules
- Ports / Network Communication Protection modules
- Addin’s and Drivers collision

Perform
- Install OS. Install Endpoint Product. Install Product under test.
- Verify conflicts in registry entries due to other Endpoint product being installed.
- Verify process behavior of Endpoint products.

Reporting
- Defect: Process crashes/restarts on a machine with other Endpoint product installed
- RAG Status - Red
Summary

- For a product most of the on field issues comes under Compatibility and hence a collaborative approach is what require to enhance the product compatibility quality.
- Compatibility is no doubt a daunting task but it installs confidence in the product quality.
- Prioritized and a framework driven execution is always better than Random pick execution.
- Ask yourself if you are Proactive or Reactive?
- Collaborate with SEs, Beta Customers and as much customer interaction to capture the software usage.
- Post release a Support~QA review of the Compatibility software metrics list only enhances your prioritization.
- No software is 100% defect free but we can make our little contribution to make it at least 99% defect free 😊
For any help/query

Ashish_Khandelwal@McAfee.com
Shishira_Rao@McAfee.com
Amrita_Desai@McAfee.com