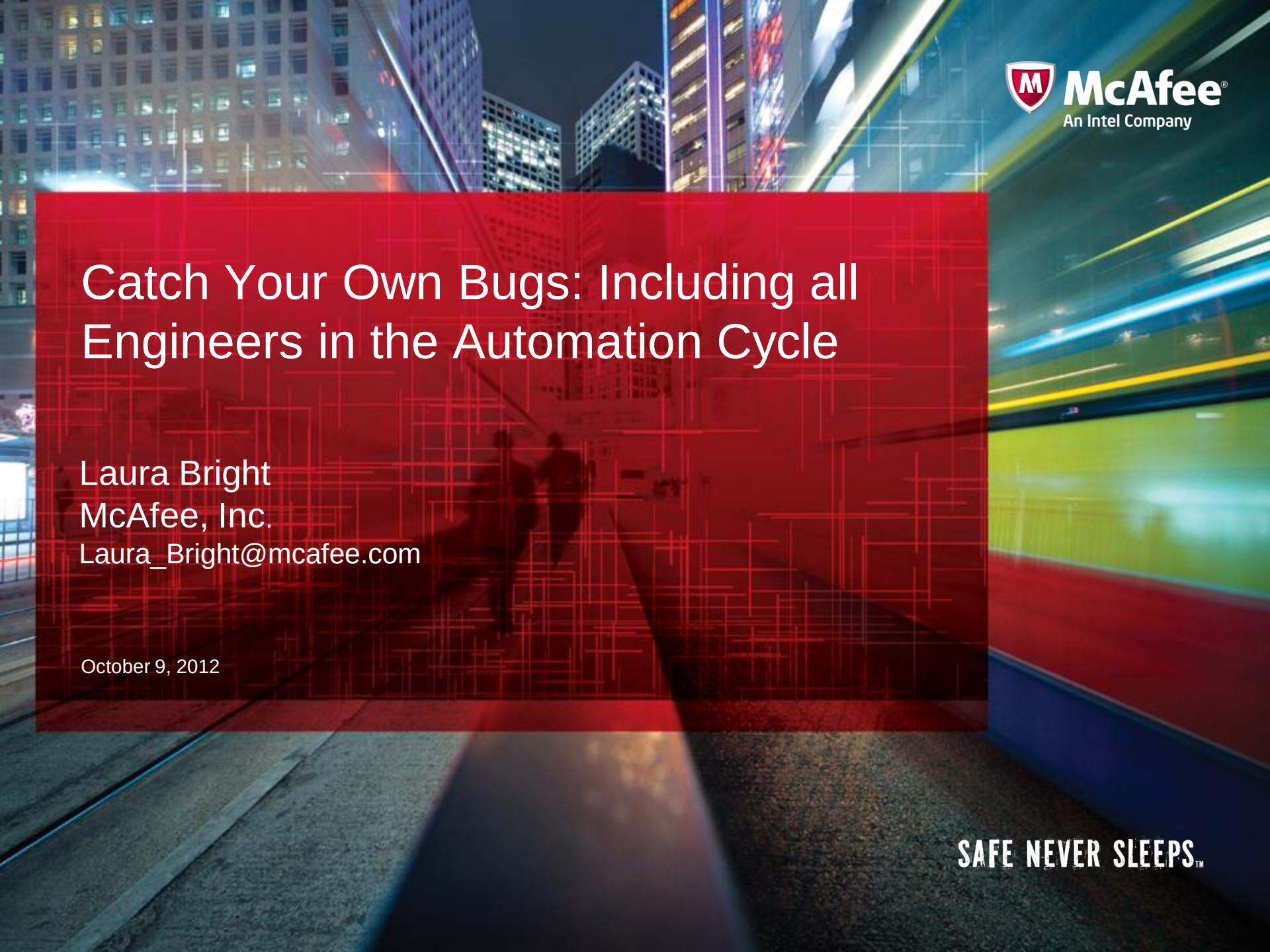


Catch Your Own Bugs: Including all Engineers in the Automation Cycle

Laura Bright
McAfee, Inc.
Laura_Bright@mcafee.com

October 9, 2012

Introduction

- End-to-end automation frameworks provide many benefits
  - Continual monitoring of product quality
  - Faster defect detection
  - Increased productivity
- Challenge: Get the entire team to monitor automation results
  - Often the automation engineers are needed to interpret test results
  - Reduces the benefits of automation
  - If intervention is required to interpret results, we're not fully automated!
- Goal: Automation framework that all developers and QA engineers can use and understand
  - Keep all team members in the loop
    - Everyone should be aware of latest results
  - Minimize overhead for automation team
    - Everyone should be able to interpret results with minimal assistance
  - Ease of writing, understanding, and maintaining test scripts

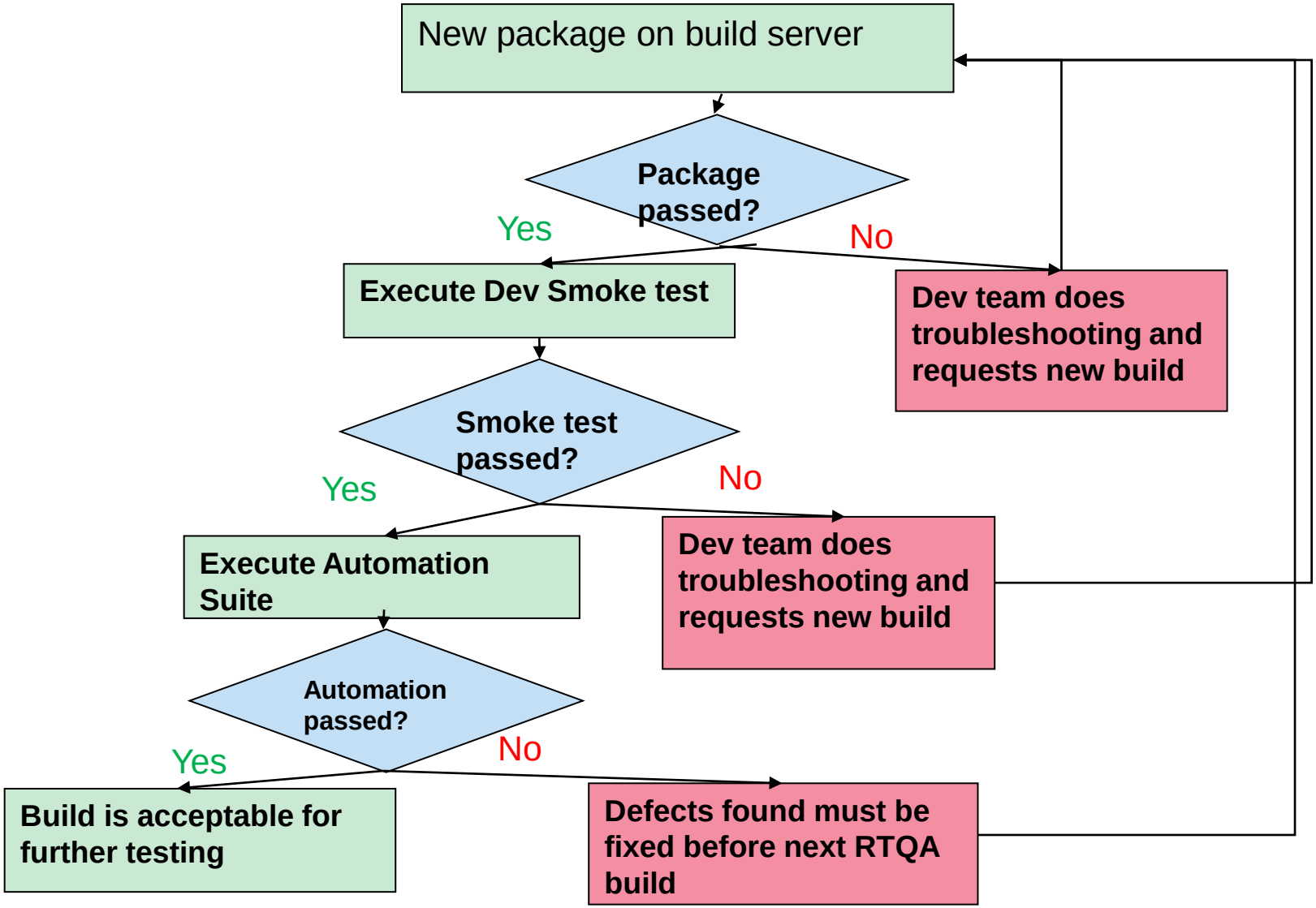
Overview

- Background: Existing processes and limitations
- Automation framework features
- Results and success stories
- Future directions

Background

- Automation for McAfee Endpoint Security Products
  - Anti-Virus and Firewall
- Geographically distributed Development and QA teams in Beaverton and Bangalore
- Build server maintained by a dedicated team
  - One automatic build per day, additional builds can be requested
  - At least one build per week marked as RTQA and used for further QA testing
  - Status of every build is tracked in a database maintained by the build server team
- Automation rigs at both sites execute tests automatically on every new build
  - Build Verification Test (BVT) suite verifies basic product functionality and runs on every build
  - Functional Verification Test (FVT) provides more in-depth coverage and runs on weekly RTQA builds

Build process flowchart



Challenges

- Frequent code changes and builds
  - High degree of code churn
  - Several builds per day
  - Manual testing cannot keep up
- Dependencies on other products
  - Interaction with other products, e.g., Anti-Virus Engine
  - Each dependent product has its own development and testing cycle
  - Changes or unknown defects in dependent product could break functionality
- Geographically distributed team
  - Development and testing occurs nearly 24 hours a day
  - Cross site interaction may be delayed
  - Changes made at one site could impact development at another site
  - Defects must be caught early to maintain cross site productivity

Limitations of Earlier Efforts

- Why don’t developers and manual testers monitor automation results?
  - Lack of time
    - Several builds a day
    - Need reminders to check automation reports
  - Difficulty interpreting results
    - Logs are difficult to read or don’t have enough information to determine root cause
    - Understanding log files may require detailed knowledge of automation framework

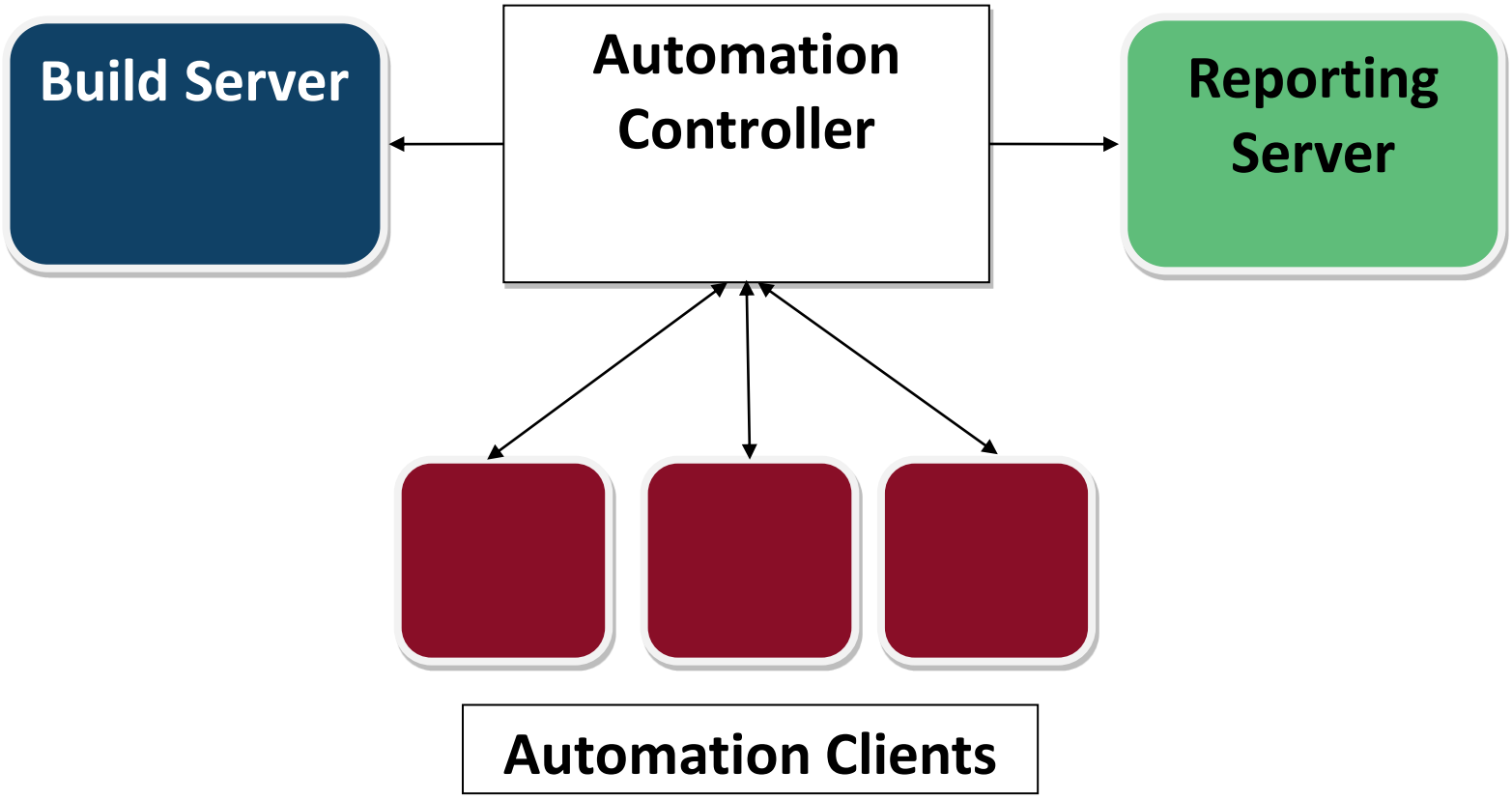
Overview

- Background: Existing processes and limitations
- Automation framework features
- Results and success stories
- Future directions

Automation Framework Overview

- Original framework built for MOVE AntiVirus (McAfee Management for Optimized Virtual Environments)
- Code base was adapted and enhanced to support Endpoint security product
  - Many core functions in end-to-end framework were reusable as-is
- Implemented and maintained by 5-6 automation engineers
- Automation efforts have full support of QA and Dev managers

Automation Framework Overview



Test Scripts and Functions

- Perl based automation framework interprets a set of test scripts
- Each script is a series of function calls
  - No detailed programming knowledge required
- Log file records each step of the test and indicates PASS or FAIL, along with relevant error messages

- Simple example: Verify that an eicar file is detected by anti-virus software and the detection is logged:

```
SetLogLocation ‘1234.txt’

CreateEicars ‘eicar.exe’

VerifyLogTextC ‘Deleted.*eicar.exe’ 1 ‘1234.txt’ 1

VerifyFileExists ‘eicar.exe’ 0
```

```
1 PASS      Log location set to 1234.txt

2 PASS      Eicar file eicar.exe successfully created

3 FAIL      Specified text ‘Deleted.*eicar.exe’ not found

4 PASS      File ‘eicar.exe’ does not exist. Expected.
```

Email Notifications

- Send an automatically generated summary of every automation run to all stakeholders at both locations (all Dev, QA, and Managers)
- All team members will be notified of results 24 hours a day
  - Automation engineers do not need to be present
- Email provides a high-level summary of failures
  - If there are any new or unexpected failures, automation reporting web server provides logs and other details for further troubleshooting

```
Results for 2K3ER2:
            0 tests passed and 1 failed
           Failed tests:  19045  ***** NOTE: Install failed on 2K3ER2! *****
Results for 2K8R2:
            115 tests passed and 7 failed
           Failed tests:  11336 8415 8416 11016 7544 7531 9580
Results for Win7x86:
            129 tests passed and 8 failed
           Failed tests:  11336 8415 8416 7544 7531 7712 11373 9580
Results for XP3:
            125 tests passed and 12 failed
           Failed tests:  11336 8415 8416 7544 11016 10726 9304 7712 9640 9560 9793 9580
```

Reporting Web Server

- Detailed automation logs to indicate the outcome of each step of a test case (PASS or FAIL)
- Graphical web interface allows users to drill down to analyze results
- **Product Logs** – All logs from the product, including install logs
- **Product configuration** – Text file containing product settings during test case execution
- **Event logs** – Windows events generated
- **Crash dumps** – If any process crashes during the run, WinDbg crash dump and registry dump are automatically copied to server
- Automatically generated documentation of framework functions

Reporting Web Server

Iteration: Iteration 5 Build: 9.0.0.1104.1

Automation summary for build: 9.0.0.1138.3		
TOTAL	137	100%
PASS	121	88%
FAIL	16	12%

▸	6982 :: *Common-Alerts=>Event Log*		PASS
▸	6984 :: *Common-Alerts=>Event Log*		PASS
▸	6985 :: *Common-Alerts=>Event Log*		PASS
▸	6994 :: *Common-Alerts=>Event Log*		PASS
▸	7342 :: *Common-Console=>Component UI*		PASS
▸	7359 :: *Common-Console=>Component UI*		PASS
▸	7510 :: *AV-OnAccessScan=>Exclusions*		PASS
▸	7520 :: *AV-OnAccessScan=>Exclusions*		PASS
▾	7531 :: *AV-OnAccessScan=>Exclusions*		FAIL
▸	Microsoft Windows Server 2003 x64 - VMware - Managed: ePO	2012-06-08 11:47:45	FAIL
▸	Microsoft Windows XP x86 - VMware - Managed: ePO	2012-06-08 15:22:05	PASS

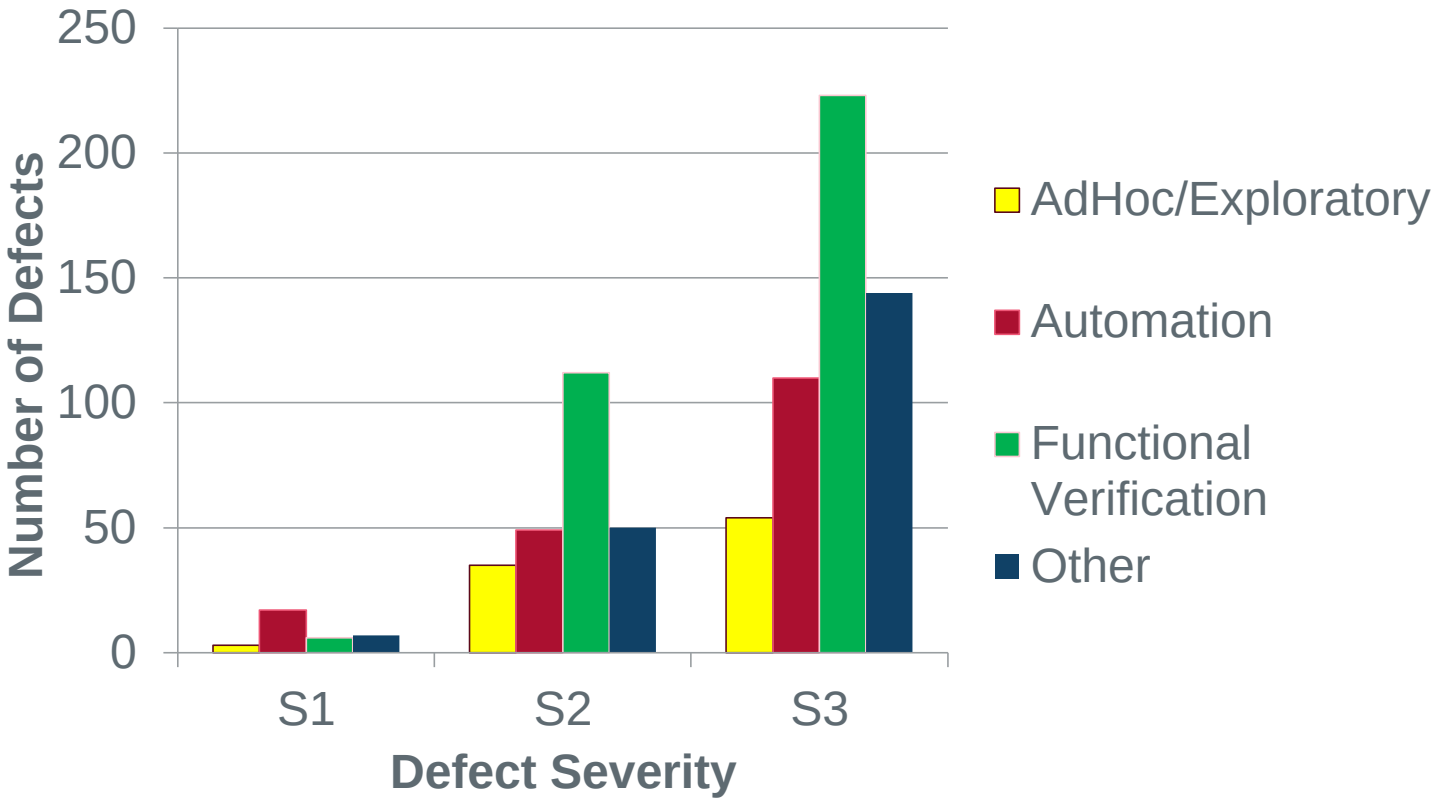
Process

- Nightly BVT runs execute stable test scripts that are expected to pass
  - Automation failures are rare and indicate new or regression issues to be addressed ASAP
- Developers can refer to web server for more information
- If additional assistance required, an automation engineer performs troubleshooting
- Continual effort to improve the information on the web server

Overview

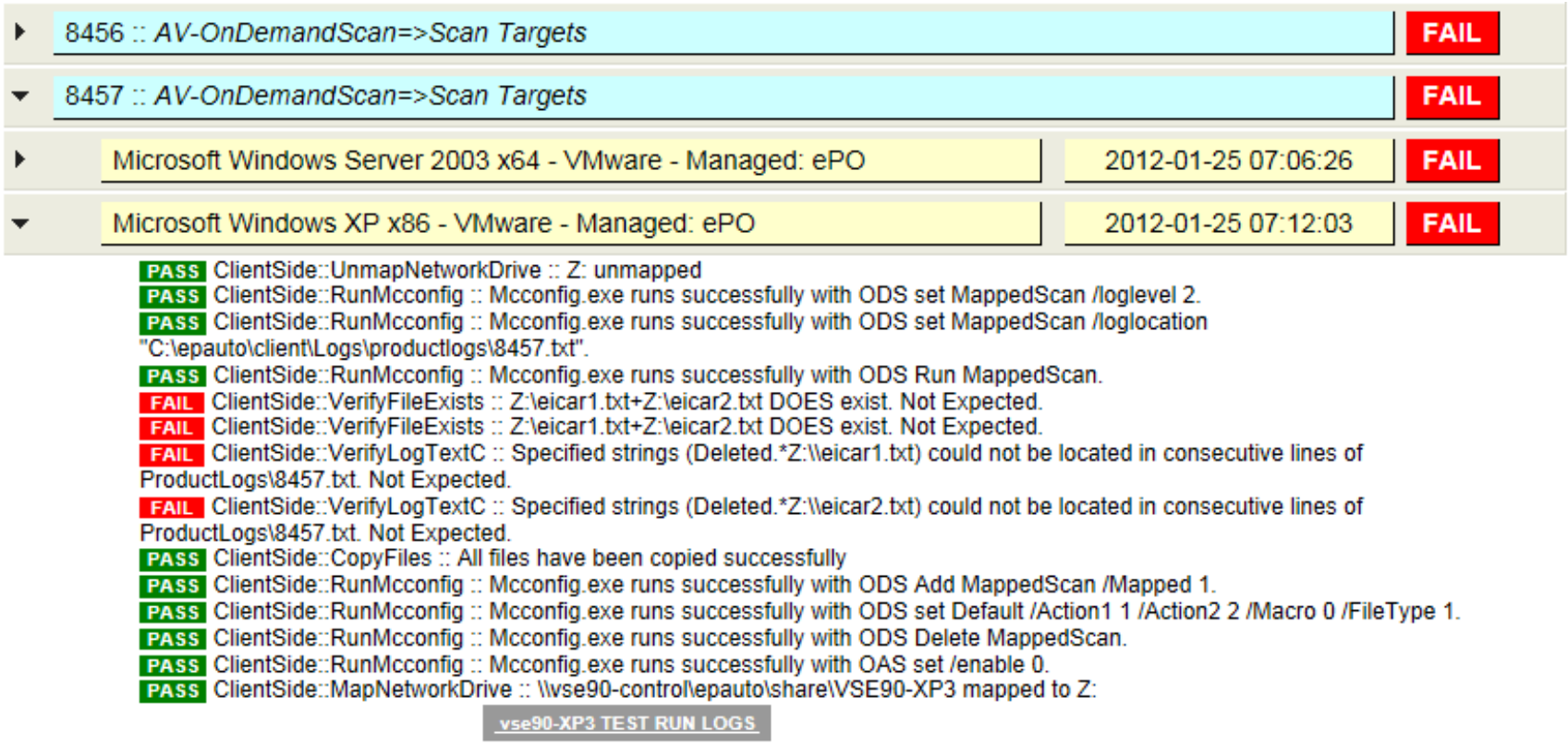
- Background: Existing processes and limitations
- Automation framework features
- Results and success stories
- Future directions

Results

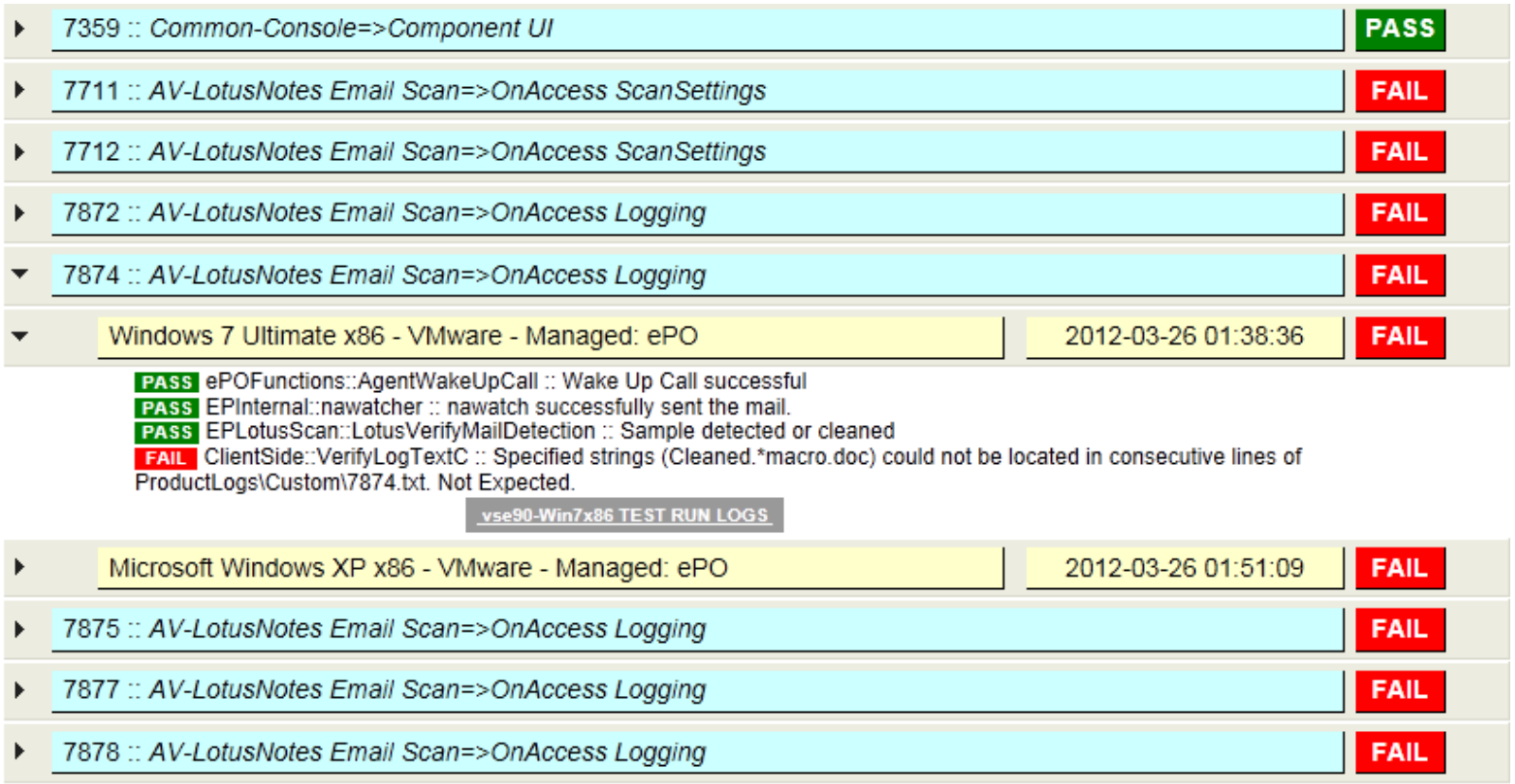


	S1	S2	S3	S4	Total
Percent defects found	51%	20%	20%	10%	19%

Regression: Failure to scan mapped drive



Regression: Email scanner logging fails



Test Scripting

- Goal: All developers and manual testers should help with test script writing
  - Increase the number of automated tests
  - Gain familiarity with the framework to make troubleshooting easier
  - Manual testers can use automation scripts to help with repetitive tasks
  - Easier to reliably reproduce defects using a script
- Bangalore Development Team
  - 5-10 test cases per developer, with help from automation engineers
  - Developers recommended fixes and improvements to framework
  - After script reviews and minor bug fixes, most scripts were added to automation run
- Beaverton QA Team
  - Initially Black Box QA team was separate from Automation team
  - All QA team members have learned automation framework and now entire team actively contributes to automation efforts

Conclusions and Future directions

- Involving entire team in automation efforts improves productivity and increases benefits of automation
- Preliminary results are encouraging
- Future Enhancements:
  - Targeted emails
  - Tracking expected/unexpected failures
    - Identify new failures after every run
    - Link known failures to bug IDs
  - Link test failures to code
  - Automation framework enhancements to Dev smoke test
  - Increase developer participation in script writing
  - Reusability for future projects

Acknowledgments

- Praveen Soraganvi, Keith Albin, Steve Nguyen, Dale Wacker, Jenny Yu, Anand Iyer, Sudhindra Kembhavi, Amit Patel
- Endpoint Development and QA Teams
- MOVE Development and QA Teams


McAfee®
An Intel Company