Releasing Software
(How do you know when you are done?)

Doug Whitney
douglas_whitney@mcafee.com

Abstract

Releasing software. How do you know when you are done? There are several items that can be added to a release checklist, but that last one may take so long that it delays shipment. Did you do the legal check? Have you completed an internal deployment? The release criteria apply to project management, program management, development, quality assurance, publications, localizations and support. It happens throughout the entire lifecycle and should not wait until the project moves into the final testing phases. There are also methodologies that go beyond the release itself, like, internal deployments, phasing out the release to a small set of customers, and product supportability. There are many more questions and we will never get it perfect, but the aim of this paper is to help you to find out what you can do to enhance the release process for you.

Biography

Doug Whitney is a development manager for one engineering team and program manager for four other teams at McAfee. He has presented papers at PNSQC and Quality Week on various topics. He has 19 years of engineering management experience and has managed QA teams at both McAfee and Intel.

Introduction

My son is a 9th grade student and had an art project that was a pencil drawing enlargement of a photograph. It was a man playing the guitar. He showed it to us and, being the proud father, I said it was wonderful. He responded, “But dad, I am not done”. He was not ready to “release” it to his teacher for a grade, even though he “tested it” on his mother and I. So I asked into this and he indicated he needed to erase some of the area on the guitar and change the background. The idea of testing a release on a small sample is nothing new. To have checkpoints along the process to validate that all necessary requirements and mandates are completed, is a necessary part of what we do. This is not just in a school art project, but also in releasing software. Here is his final picture, not bad for 14.

This paper is about the process we use at McAfee to release software. It will touch on the checklists we use when releasing, the entrance and exit criteria we use for milestones, and the idea of phasing out the software both internally and externally.

The checklists

At each milestone, there is a checklist that validates what needs to be accomplished prior to moving on to the next one. We also have a milestone checklist for the planning processes we use at McAfee. Our product lifecycle framework (PLF) spreadsheet has different tabs for each of the milestones.
Items to consider at each of the milestones ***that can influence*** the release:

Plan commit milestone checklist:

- Identify any legal requirements for 3rd party software being used
- Milestone release criteria established
- Identify software and hardware requirements and timelines for acquiring
- Localization plan reviewed and scheduled
- Risks identified with mitigation steps
- Testing and defect status reporting established
- Rollout schedule created (Security as a Service - SaaS)

Code complete milestone checklist:

- Sign off from legal for requirements
- All required software and hardware is available
- Localized drop completed
- Internal deployment requirements reviewed and approved
- Release to Support plan initiated (corporate)
- Testing and defect status report presented weekly
- Security Audit complete

Release candidate milestone checklist:

- Bill of materials created
- Code analysis completed
- All required external approvals received (Windows Hardware Quality Lab (WHQL), Win Logo certification, etc)
- Product compatibility testing completed
- Live Virus/GOAT (sacrificial goat – think Jurassic Park) testing completed
- Readme file sent to pubs for final review and localization
- Release to Support plan completed and reviewed
- Supportability document draft created
- Performance / Soak tests completed and results meet expected gains

McAfee has made a strong effort to create a comprehensive checklist that will reflect what needs to be completed at each milestone. I have attached the actual checklist items in the appendix. The checklists are broken down by each functional group and have sections for product management, program management, development, QA, technical publications, support and localization. Not all groups have activities at each milestone, but some have activities in all of them.

Testing Activities by PLF Phase & Milestones

Iteration Testing

Goals

The goal of this testing is to validate that the functionality of the feature delivered as part of the iteration. It validates that the base level of functionality defined for this application is stable and that new functionality added for this release has not destabilized the product.

This test pass will:

1. Verify that the product's pre-defined Build Verification Test (BVT) passes with the "Iteration complete" build.
2. Verify that the product is stable enough to support more detailed testing.

Entry Criteria

1. The development manager declares that a "Iteration complete" build for the iteration is available. In this build, all new or modified features and functionality specified for this release are fully implemented and ready for testing.
2. Any bug fixes or enhancement requests targeted for this release, and specified within the release planning documents are fixed and included in the 'feature complete" build.
3. The "iteration complete" build is available to QA as an official build.
4. The BVT for this product is updated based on delivered functionality
5. The Bugzilla Defect database has been setup for this product.

Exit Criteria

1. All tests within the product's BVT pass
2. All test plans required for the iteration testing phase of the project are complete, reviewed, and approved.
3. All detected defects have been logged in Bugzilla.

Beta Testing

Goals

The goals of this test pass are to demonstrate an increased level of stability as defects found in the previous test passes are fixed, and to increase the level of compatibility testing by including secondary languages and platforms in the test cycle.

This test pass includes:

1. A complete execution of functional test plans associated with each area of the product.
2. A complete regression test of all current test plans
3. A complete pass through all defined integration tests.
4. Compatibility testing of primary supported languages.
5. Conduct ad-hoc testing will be performed on application
6. Focus on resolving beta feedback issues
7. Completion of Live Virus Testing.

Entry Criteria

1. The schedule and test plans have been reviewed to determine if, based on the results of the previous test pass, the testing planned for this pass should be increased or reduced.
2. The project team has agreed upon any required changes to the testing for this pass and the schedules have been adjusted accordingly.
3. Development has executed a Build Acceptance Test (BAT) and has unit tested all bug fixes against the Release.
4. Development of all required test plans and procedures for this test pass is complete.

Exit Criteria

1. All planned tests have been run.
2. All detected defects have been logged in Bugzilla and scrubbed
3. No test blocking bugs exist which would prevent entry into the next test pass
4. All "Beta Critical" defects have been resolved.
5. All fixed defects have been included in an official build and verified.
6. All Severity 1 defects fixed in the release to date have been re-verified, and found to be fixed, in the Beta candidate build.
7. Pseudo-translation builds to be tested (Internationalization -I18n)

RC (Released Candidate) Testing

Goals

The goals of this test pass are to complete the planned testing for this release. At the end of this test pass every planned test will have been run on at least one Operating System (O/S) or platform combination.

This test pass includes:

1. Focus on regressing previously discover defects and issues
2. Conduct ad-hoc testing will be performed on application
3. Successful completion of Live Virus Testing.
4. Compatibility testing of primary supported languages.
5. Continued testing of various O/S and languages.

Entry Criteria

1. All beta feedback has been incorporated into the product or otherwise responded to.
2. All Severity1 and 2 issues have been resolved.
3. Development has executed a Smoke Test and has unit tested all bug fixes against the Release.

Exit Criteria

1. All planned test plan have been executed.
2. The product has been tested all planned platform/browser/language combinations.
3. All detected defects have been logged in Bugzilla and scrubbed.
4. All “Release Critical” defects have been fixed, verified, and closed.
5. All fixed defects targeted for this release have been regressed.
6. Fixes for all severity 1, and priority 1 defects fixed in this release have been re-verified with the RC build.
7. Incoming defect rates have been declining for at least 2 weeks prior to this milestone.
8. Key beta testers agree that the product is ready to ship.
9. Product has successfully passed Live Virus Testing recognized by McAfee Labs

RTW (Release to World) Testing

Goals

The goals of this test pass are a final validation of the product. This is primarily to verify any fixes included in the RC candidate build, and to ensure that those fixes have not destabilized the product.

This test pass includes:

1. Final validations of the RC candidate build.
2. Verifying and closing any defect fixes made in the RC candidate build.

Entry Criteria

1. An official RC candidate build has been created that includes all defect fixes targeted for this release (i.e. no unfixed, release critical, defects remain).
2. Development has executed an IBT and has unit tested all bug fixes against the Release.
3. Release documentation has been created, reviewed and approved by QA

Exit Criteria

1. All code changes made since the RC milestone have been inspected by a cross-functional team.
2. All defects fixed in this build have been verified and closed.
3. All testing targeted for this release is complete with summary reports available.
4. Product release notes have been inspected.
5. Agreement from all functional groups that the product is ready to be released.
6. Release Plan, Process, and Schedule created and ready for use
7. Zero article testing (download and install as a customer would do) is complete and passed to the Deployment and Integration (DI) Team)

RTM (Release to Manufacturing) Testing

Goals

The goals for the RTM phase is to complete the validation of the final RTW build as it is integrated into the retail CDs.

This test pass includes:

1. Validation of the CD images and compliance to RTM Bill of Materials (BOM).
2. Release of Master CD's to manufacturing.
3. First Article Inspection completed.

Entry Criteria

1. Complete and signed off RTM BOM.
2. No product fixes or changes made to the build included in image.
3. ISO images pass initial Acceptance test.

Exit Criteria

1. CD images meet the requirements established in the RTM BOM.
2. 100% of planned testing competed.

No trade mark violations and correct documentation and PP included on CDs.

Phasing out the release

McAfee in Action

The internal deployment of enterprise products within McAfee is referred to as “McAfee in Action” (MIA). This is not to be mixed up with McAfee inaction. ☺ McAfee releases its software internally through a series of deployments that are geographically based. We have offices that conduct these deployments in Beaverton, Santa Clara and Plano Texas. There is a MIA coordinator who manages the expectation of the IT groups, who actually do the deployments, as well as manages the schedule for rollouts. This type of deployment allows for the build to be consumed by a professional IT organization prior to deploying to the world.

The phased rollout

The phased rollout approach is where we upgrade a subset of customers and wait to see if there are issues. We use this approach in several ways depending on the type of software.

Enterprise
For our enterprise customers, we have a process in our lifecycle called RTS, or release to support. The support engineer then will work with our development engineers to release to a subset of customers we refer to as the early adopters. We have goals for Early Adopter Program (EAP) that are product specific. Some products will want to roll to 100 customers and others will want to roll to 100,000 total nodes. Engineers will often go on-site with these customers to ensure that any rollout issues are understood and documented. Once the issues are documented and any critical ones addressed, we will make the software generally available by moving it to the download site for remaining customers.

SaaS model
McAfee has had SaaS model software offerings for over 9 years. The phased rollout approach we use is based on configurable settings on the NOC (Network Operating Center) that can be modified to upgrade based on the customer identifier in our database. We will start with all new installs and by identifying customers that will provide us with a total of 5000 nodes. We usually find ones that do not have custom policy, or may be on an OS platform we have supported for some time. Once we have the customers identified and they rollout, we stop and determine if there are any calls to support. We also use a method where the installer will report success or failure during the install process and upload a small response to the NOC. We can then query this data for success or failure. The install status reporting is a switch that can be configured since we do not need this operation running during normal operation, only during the phased upgrade process.

The phases are established and documented in a phase rollout plan that is part of the initial planning documents we use to get to a plan commit status in our product lifecycle. We start with all new customers. We then roll out to 5,000 upgrades, and then a week later to 15K more. We then take a break for about a week and assess issues. If there are no issues that need to be addressed, we will move forward with 60K, then 125K, then 250K and then open it up for the remaining customers. We also plan an update to the code and schedule it as a potential production refresh.

Consumer
The consumer team approaches things differently. They start with Australia. The release of software starting with Australia actually provides many advantages.

- They are in a time zone that is close to one of our larger development center in Bangalore.
- The development center and call center in Bangalore are in close proximity
- It is a statistically valid sample
- They speak English in Australia

At the call center, we will have engineers on site during the rollout. If there is an issue raised on the phone that is not known, the engineer will listen in on the call and determine if there is additional action or information needed. In all cases we manage the information received and stop the rollout if critical issues are discovered. Once fixed, we begin the cycle again.

Conclusion

When the areas that are time sensitive are scheduled early and acted on, it reduces the risk of a late deliverable. When each milestone has clearly defined entrance and release criteria, it will reduce the risk of items being carried over to the next milestone (or released). When the internal deployment or a phase release discovers an issue, it reduces the risk of it being seen broadly in the field. These are approaches that can be used to help answer the question, “How do you know when you are done”.

Appendix

Example of SaaS rollout in table format:

Partner	Languages	RTQA	RTW (Go Live Date Client & NOC)	Comments
Pre-Phase 1				
McAfee - Beta	ENU		5/21/2009	ENU Only
Beta Refresh	ENU	7/14/2009	7/16/2009	RTW bits deployed to Beta
Rollout Phase 1				
McAfee - Phase I-A	All	-	7/7/2009	All "New Installs"
McAfee - Phase I-B	All	-	7/14/2009	5K English only - no custom policy - No Browser Protection
McAfee - Phase I-C	All		7/16/2009	15K English only - no custom policy - No Browser Protection
Potential Production Refresh	All			Production refresh deployed
McAfee - Phase I-D	All		8/20/2009	Restart of the Rollout 20K English only - no custom policy - OK Browser Protection
McAfee - Phase I-E	All		8/27/2009	60K
	All		9/22/2009	125K
McAfee - Phase II	All	-	9/24/2009	260K
McAfee - Phase III	All	-	9/29/2009	Remaining users

Concept Commit Checklist

Function	Item or Deliverable
Product Mgmt	
	Present any critical Feature Modification Requests (FMR), new platform support / service pack or 3rd party compatibility testing
Program Mgmt	
	Patch Milestones created to be tracked in Health Checks and Plan of Record (POR) meetings
Development	
	Consumed component team has been made aware of the timelines and there are no issues
	Engineering Statement of Work (SOW) - DEV estimate complete on high, medium and low defects
	Initial SOW complete (identify priority 1,2 and 3 defects for inclusion)
QA	
	Engineering SOW - QA estimate on backend schedule with a quarter level timeline for Release to Support (RTS)
	Initial SOW complete (identify priority 1,2 and 3 defects for inclusion)
Tech. Pub's	
	Commit from Tech Pubs for Readme Review / Sign off
Localization	
	Engineering SOW reviewed by Localization
Support	
	Agreement on prioritized SOW
	Top customer issues have been identified

Plan Commit Checklist:

Function	Item or Deliverable
Product Mgmt	
	Schedule reviewed & approved
Program Mgmt	
	Identify any legal requirements (3rd party exports, etc.)
	Master Schedule reviewed & approved
	Milestone Review Plan completed for the project. This plan consists of a copy of this spreadsheet with the milestone criteria tailored to suit the specific project
	Milestone Review Plan completed for the project. This plan consists of a copy of this spreadsheet with the milestone criteria tailored to suit the specific project
	Project entered into Bullseye
	Project Launch Meeting with full team
	Risk Analysis Matrix
	Risk List Generated for weekly tracking
Development	
	Build System Setup to start development and coding (e.g. SVN repository, Branching Plan, Team City etc.)
	Development Schedule reviewed & approved & communicated
	Development Plan reviewed & approved
	Functional Spec's or equivalent reviewed & approved (If using Sustaining Agile specify the Iteration Plan if applicable, i.e. number, duration, etc)
	Integration Plans Created and signed off between product teams (common, shared components: VSCore, McAfee Foundation Services (MFS), HIPSCore, etc.)
	Merge plan draft has been completed to identify any mainline forward merges
	Post Concept Commit requirements qualified
	Required hardware identified and ordered
	Required software identified and ordered
	Required training identified and scheduled
QA	
	Automation Plan has been created, approved and resourced
	Bugzilla product & version created & accessible
	Content Security Testing (Host Intrusion Prevention (HIP) Content, Live Virus Testing (LVT) and GOAT testing approved)
	Project entered into Bullseye with key milestones setup, risks, dependencies
	QA Plan reviewed & approved & communicated

	Required hardware identified and ordered
	Required software identified and ordered
	Required training identified and scheduled
	Send out notification all parties on schedule and plan
	Test Case DB setup and Functional Decomposition updated
Localization	
	Localization Plan reviewed & approved
	Localization Schedule reviewed & approved
Support	
	Schedule reviewed and approved
	SOW plan has been updated on tier 3 central

Feature Complete Checklist:

Function	Item or Deliverable
Program Mgmt	
	Sign off from legal on any 3rd party inclusion
Development	
	All P1's Customer Fixes implemented and any deltas communicated to support and Product Management
	Drop to Localization team
	Feature complete build (BVT Passed) delivered with Software BOM & release notes
	Identify defects for merging
	Software BOM created
	Unit tests complete and passing, including report on coverage
QA	
	All reported defects have been scrubbed in Bugzilla
	All required HW & SW is available
	All required test scripts reviewed, and approved
	BOM validated by QA
	Bullseye status is being updated at least weekly
	BVT Passing 100%
	Determine what content should be in the readme
	Install / Deployments tests complete
	McAfee In Action Initiation Document reviewed and approved
	Scalability lab testing complete, results verified as expected
	Testing and Defect Status Report presented weekly (e.g. Tiki, Twiki)
Support	
	RTS draft plan is created

Release Candidate Checklist:

Function	Item or Deliverable
Development	
	All static analysis reported bugs have been reviewed & responded to
	All deferred defects have been reviewed by the cross-functional team (Sustaining DEV & QA, Support, PM)
	All fixes that are common with current mainline code base have been merged per Standard Merge Process
	All open defects have been fixed or deferred
	Complete list of included common McAfee components (e.g. Engine, Agent, Virus Scan Core) has been created. Included versions have been confirmed as releasable by component team
	Patch Supportability Document Draft created
	RC build delivered with Software BOM updated & release notes
	Source code escrowed
	Unit tests complete and passing, including report on coverage
QA	
	All Bugs Verified and Smoke / Stability tests performed around components affected; Bugs verified on at least 2 Platforms
	All planned testing complete with results recorded and 100% pass rate. Any failures have associated defects logged in Bugzilla, with the defects marked as "deferred"
	Any required external approvals have been received (e.g. Windows Logo Certification)
	Bullseye status is being updated at least weekly
	Compatibility testing with other McAfee products complete. This includes testing will all products sold in a suite with this product. (full Point Product Compatibility test (PPCT) testing, i.e. ePO, McAfee Agent)
	Exploratory and Area testing completed around affected components
	Final package has passed an complete in-depth Extended Build Validation Test (eBVT)
	Final package marked in eCM with milestone
	Final package validated against the BOM
	Goat testing passed with results approved by Avert (AV Scanner products only)
	Install/Uninstall testing confirms a complete uninstall (verified with "beyond compare" or similar tool)
	LVT passed with results approved by Avert (AV Scanner products only)
	Patch tested in EAC setup
	Performance / Soak tests completed and results meet expected gains (equal to or better than previous patch depending on code changes)
	RC Build sent to MVT team
	Readme draft sent to tech pub for final review

Tech. Pub's	
	Readme's are localized
Support	
	RTS Phased plan completed and reviewed by team

Release to Support Checklist:

Function	Item or Deliverable
Program Mgmt	
	All sustaining PLF milestones reviewed and complete
Development	
	All static analysis reported bugs have been reviewed & responded to
	All fixes that are common with current mainline code base have been merged per Standard Merge Process
	All Priority 1 and reopen defects are resolved
	Any defects fixed since RC have received a detailed cross-functional code walkthrough and review of appropriate testing (DEV, QA, Support)
	Any MIA ship stopping defects have been fixed and unit tested
	Appropriate deferred defects identified and move to next patch or mainline release
	RTS build delivered with updated Software BOM & release notes
	Supportability initial draft reviewed
	T3 Support training completed as needed
	Unit tests complete and passing, including report on coverage
QA	
	All escalation fixes merged into mainline have a defect created for current mainline release
	All fixed defects are verified & closed
	All RTS builds have been promoted to RTS status within the build system
	All site deployments planned and at least 80% successful install with no showstopper issues
	All testing defined in test plan is complete with any failures approved
	Any open/deferred defects are scrubbed and disposition complete
	Bullseye status is being updated at least weekly
	Final package has passed an complete in-depth eBVT
	Final package marked in eCM with milestone
	Final package validated against the QA BOM
	Final readme reviewed and packaging complete
	Final report is updated in Twiki or central location
	Install/Uninstall testing confirms a complete uninstall (verified with "beyond compare" or similar tool)
	LVT passed with results approved and signed off by McAfee Labs
	MIA Closed, with all reported issues fixed and verified or deferred
	MIA Sign Off (IT, MIA Admin)
	Non-English builds are fully tested and ready for RTS

	Performance figures validated on release build without regressions (If % improvement expected then this should also have been met)
	RTS build has been tested against McAfee Virtual Technician (MVT) and passes
Tech. Pub's	
	Patch readme has been reviewed and signed off by tech pubs
Localization	
	Documentation complete & localized as required (including readme & release notes)
Support	
	Any updates made to RTS plan based on MIA or final testing
	Addendum Release notes Knowledge Base (KB) is ready

Release to the World Checklist:

Function	Item or Deliverable
Program Mgmt	
	All PLF milestones updated
	Retrospective scheduled for the project
Development	
	Any ship stopping defects found during RTS have been fixed and unit tested
	RTW build delivered with Software BOM & release notes
	Supportability document completed, reviewed, & approved
QA	
	All release packages have been posted to back up server for AV testing and storage
	All RTW builds have been promoted to RTW status within the build system
	Any multi-langauge testing which was not completed at RTS should be finalized
	Any ship stopping defects found during RTS have been verified
	Internal tracking tool (Bulls Eye) updated with final status for patch project
	Final package has been posted to the download site & download testing is complete
	Final package marked in eCM with milestone
	Final package validated against the BOM
Support	
	Readme KB addendum has been updated with any issues found in RTS
	RTS Phased release completed and no ship stopping issues found (at minimum 80% deployment with 100% success rate)