

How Automation Closes the **PATCHING GAP**



PUBLISHED BY

GREG CHARMAN

KELVERION VP OF SOLUTIONS & SERVICES

Contents

INTRODUCTION	3
TROUBLE WITH REBOOTS, PART 1: EVERY SERVER AT THE SAME TIME.....	4
TROUBLE WITH REBOOTS, PART 2: NEW SERVERS RANDOMLY REBOOTING	6
IN CONCLUSION	13

INTRODUCTION

Monthly patch deployments of software and security updates can be a costly, time consuming, and unreliable process, leaving companies with huge security and compliance issues. Many tasks are repeated monthly and it is up to the SCCM Administrator to ensure that devices are patched correctly and working. When the Admin is looking after a large estate or multiple customers, the patching process often becomes an unwieldy full-time job fraught with failure gaps.

This White Paper will first focus on locating those gaps and then offer the automation solution that can tighten the process.



TROUBLE WITH REBOOTS, PART 1:

EVERY SERVER AT THE SAME TIME

In isolation, a device reboot is fine, but when the machine is a server supporting a production service, it is essential that its reboot be considered as part of the wider service estate.

For example, if a service has a Web front-end made up of three IIS Web servers, and all three reboot at the same time, the Web front-end would go offline and the service would be unavailable. It is therefore essential not to deploy patches to all three machines at the same time.

REDUNDANCY AND STAGGERING REBOOTS

The importance of redundancy is without question, particularly for high availability solutions like online front-ends, database servers, and communication platforms. However, for redundancy to work, each system involved must be able to take on the functions of the other, which also means that the patching process for each is likely to be similar or even identical.

As patching may require reboots or other cases of downtime, it is essential that individual patching processes are staggered to avoid a loss of service level.

In a complex application with multiple layers - presentation layer (top), processing layer (middle), database layer (back-end) - it is often the case that each layer is entirely dependent on the tier below it for correct operation. This interdependence is often unable to handle even a temporary loss of a lower level during a reboot. Therefore, each tier must be patched at a different time, starting with the back-end first, followed by the middle, and finally the top layer. Patching and rebooting each layer in this order results in the application returning to full functionality.



Patching upward from lower layers keeps applications operating.

DEVICE COLLECTIONS AND MANUAL PATCHING

IT departments often opt to create a number of Device Collections in SCCM and patch the machines in each collection at different times. Yet it is nearly impossible for the Admin to know - for every case - which devices should be in a particular patch schedule. The Application Developers and Infrastructure Support teams are those who best understand the applications and their optimum patching cycles. However, they are rarely familiar with patching schedules and can only define which devices go into a schedule through lengthy communication with the SCCM Admin on an application-by-application basis.

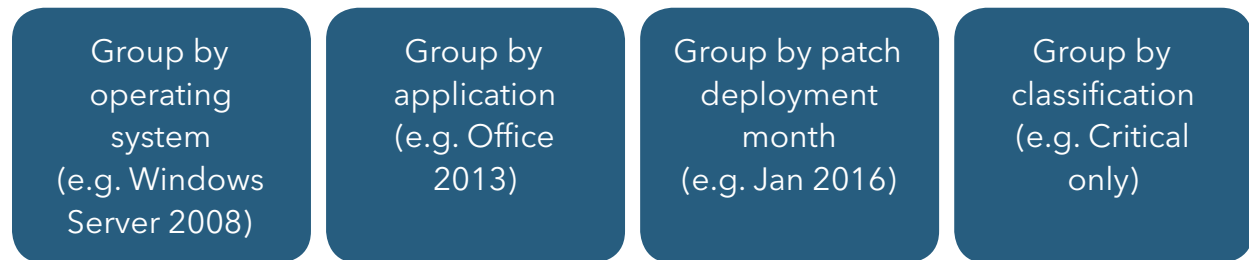
Further, some essential devices must be manually patched, yet there is no guarantee that those manual tasks are actually getting done every month. It is all too easy to lose track of which boxes are being manually patched, which manually rebooted, and who owns those deployments and reboots.



TROUBLE WITH REBOOTS, PART 2:

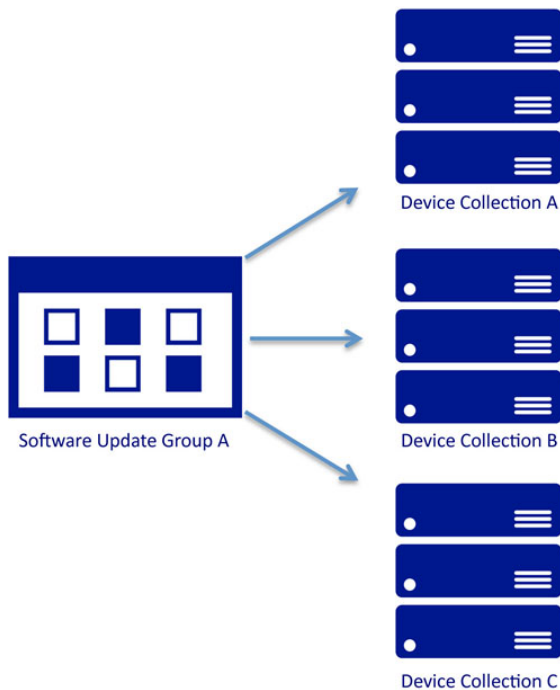
NEW SERVERS RANDOMLY REBOOTING

As new patches are released each month, the SCCM Admin assesses those patches and creates a series of Software Update Groups that bundle together sets of patches. The contents of each Software Update Group will vary depending on the patching policy of a customer. Some examples of possible patching policies include:

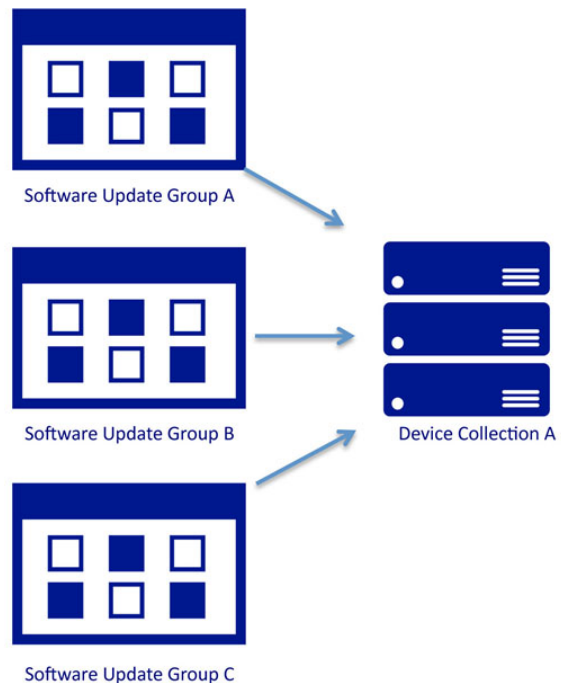


Or most likely, a combination of more than one of the above.

A Software Update Group can be deployed to multiple Device Collections in SCCM.



Multiple Software Update Groups can be deployed to a single Device Collection.



Over time, each Device Collection in SCCM is associated with a number of Software Update Groups, each containing a set of patches actively awaiting deployment to a device.

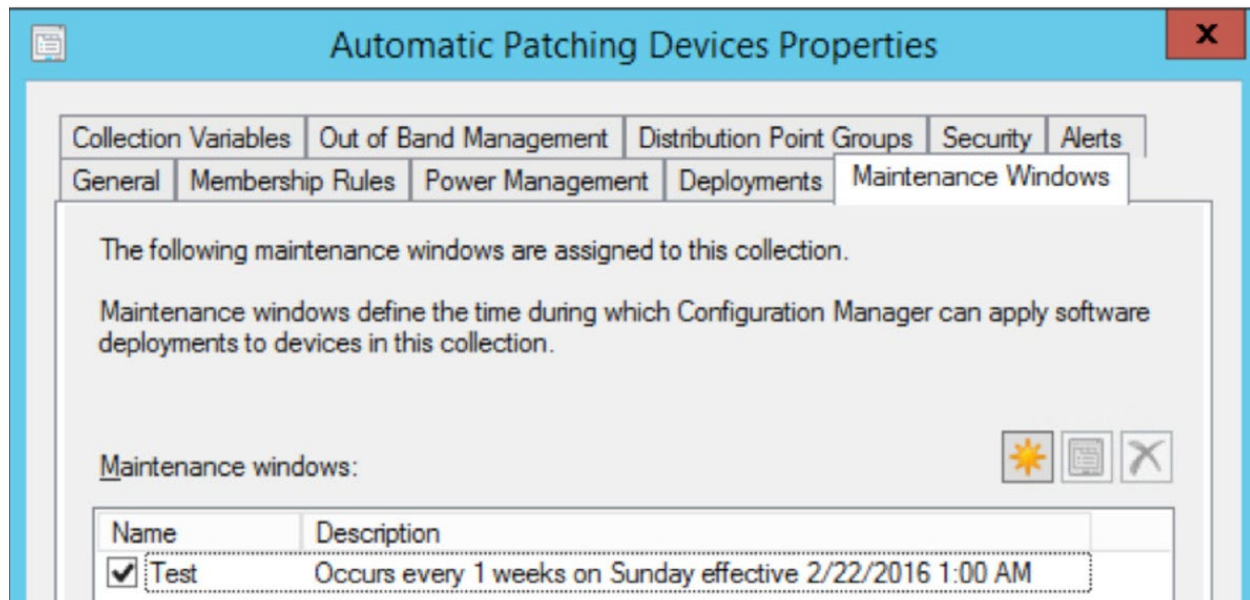
When a new server is introduced, it is added to an existing Device Collection in SCCM. The SCCM Agent on the server will periodically check policy to see if there is anything to do, this includes Active Software Update Deployments (patches). As soon as the Agent detects a Software Update Group, it will start comparing the contents to its deployed patches to determine if there are any missing patches.

Each Software Update Group will have a defined patch deployment date; this may be a date in the future, in which case the Agent notes the patch and does nothing until the deployment date. However, when the Agent detects a patch from an older Software Update Group is missing, it will immediately begin installing the patch and then reboot the new server. This will keep happening until all missing patches have been processed.

From a Service Management point of view it appears that a new server has been added to the estate, which then starts randomly rebooting. This same process of detection, installation and rebooting can also occur when a server is moved from one SCCM Device Collection to another.

MAINTENANCE WINDOWS

SCCM provides a feature that is intended to control and prevent random reboots from happening called Maintenance Window. This feature is available on each Device Collection, and it allows the Admin to set a period of time during which devices within the collection may install applications (via task sequences) and software updates.



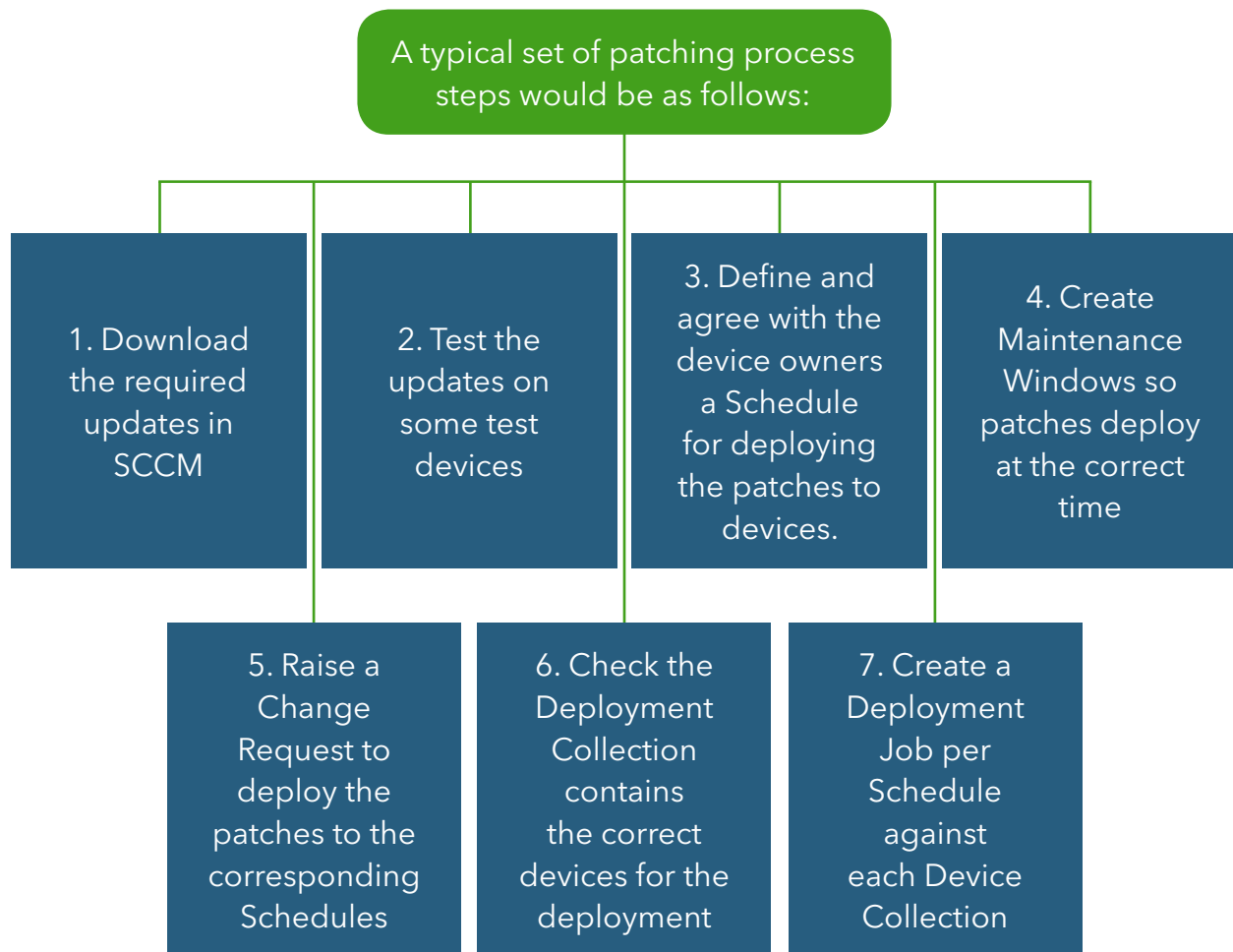
Therefore, when creating a short Maintenance Window, the SCCM Admin needs to edit individual software updates to reduce their default maximum runtime. On the other hand, a larger Maintenance Window that allows more patch variance can lead to random rebooting at unexpected times.

Another major issue faced by SCCM Admins is finding out which Device Collections have Maintenance Windows, as there is no default view or setting to display this in the

SCCM console. This can prove problematic as a device can be a member of more than one collection, and could easily have more than one Maintenance Window associated with it. This device would behave differently than one in just a single collection, and is more difficult to troubleshoot when a patch will deploy.

With all these considerations, using Maintenance Windows to control unnecessary reboots on servers requires a considerable amount of planning and administrative effort.

THE ONEROUS TASK OF PATCHING



It is a massive piece of work configuring all these steps, and it is the SCCM Admin who must ensure that devices are patched correctly and working.

MANAGING ERRORS

After requesting that SCCM deploy some patches, it is crucial to confirm which deployments were successful. Admins must typically rely on Patch Deployment Views and Reports in SCCM to determine which updates have failed. These are not instantaneous updates and are intended to confirm compliance rather than ascertain

which devices have failed patch deployment. While SCCM has a built-in logic for this, it can be time consuming to go through logs to find errors and, when found, re-do the patching.

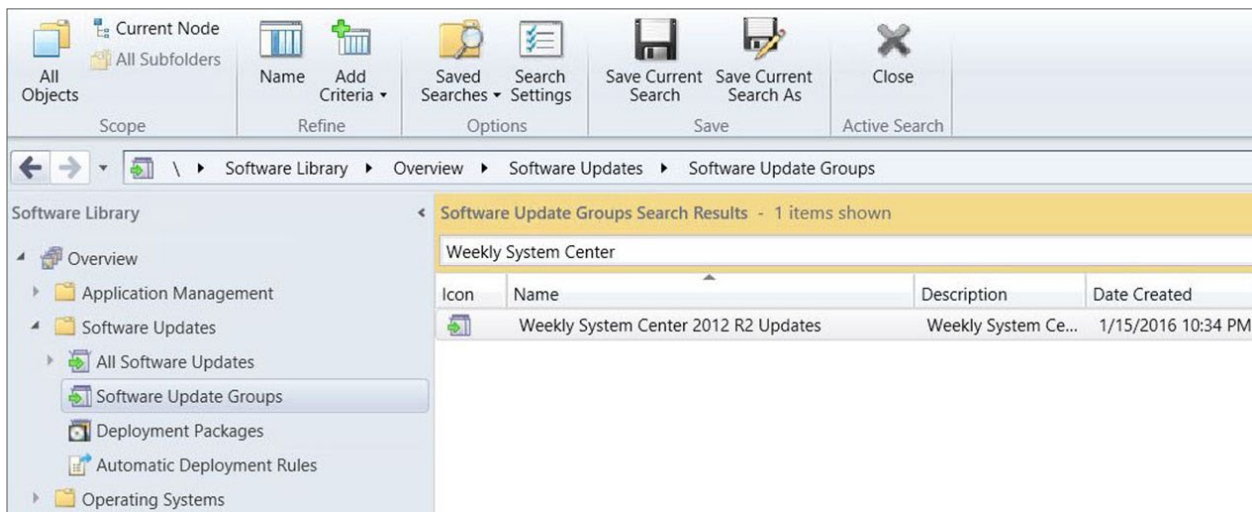
One option is for SCCM to send an error message to System Center Operations Manager (SCOM), a monitoring tool, which then dispatches an alert to the Admin. This function is not enabled by default, however, so it would need to be set up manually.

CLOSING THE GAPS WITH AUTOMATED PATCHING SOLUTION

Using Orchestrator with Kelverion’s Automated Patching Solution distills the hands-on administration of the patching process to these three simple steps:

1. Download the required updates in SCCM
2. Test the updates on some test devices
3. Raise a Change Request via the Service Desk portal to deploy the patches

Once a Change Request is approved, the Solution calculates the best date for each schedule and automatically creates deployment jobs for each Device Collection in SCCM. Rather than using Maintenance Windows, Orchestrator is used to control when patches become active and also to de-activate them after a defined period. This patch availability control by Orchestrator prevents new servers from detecting and deploying older patches outside of agreed patch windows.



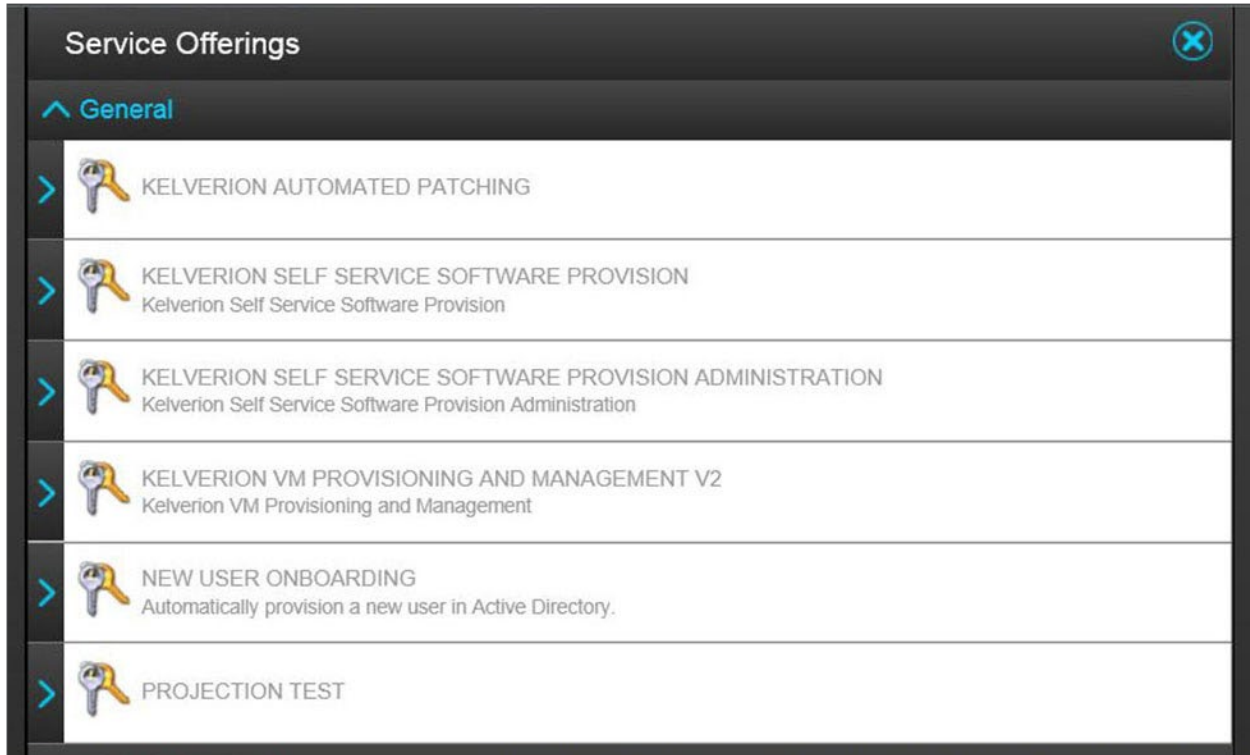
The Weekly System Center 2012 R2 Updates have been downloaded, ready to deploy.

This also allows a server to be moved between SCCM patching Device Collections at any time during the month without risk that it will install missing patches before the next agreed patch change window.

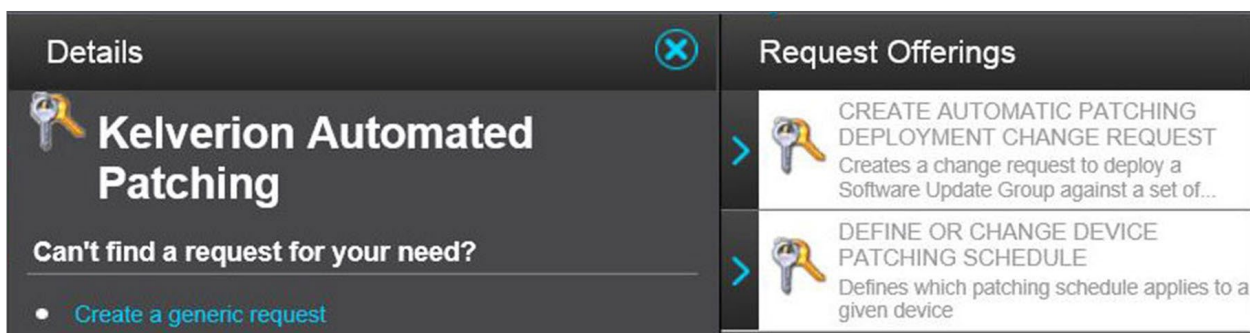
Linking deployment to a Change Request allows greater control of when the SCCM patch deployments are enabled, thus preventing unnecessary reboots of critical systems outside of an approved change control window.

FREEING THE ADMIN

With this simplified and automated Solution, the patching process can be turned over to Device Owners (SQL team, Server Support, Application Developers, etc.) to define and control. This puts things in the hands of those who best know their systems' patching needs and when their devices would be available for updating, therefore increasing service availability.





Select Automated Patching from the Service Offerings in the Self Service portal.




Select Automated Patching from the Service Offerings in the Self Service portal.



Select the Software Update Group to deploy.

*  

Software Update Group Name

 Weekly System Center 2012 R2 Updates

Select the Schedule(s) to deploy too.

*  

	Schedule Name	Reboot	Deployment Start Day	Deployment Start Time	Deployment End Day	Deployment End Time
<input type="checkbox"/>	Manual Patching Tuesday 18:00	Suppress Reboot	Tuesday	18:00	Tuesday	22:00
<input checked="" type="checkbox"/>	Saturday 18:00 Allow Reboot	Allow Reboot	Saturday	18:00	Sunday	04:00
<input type="checkbox"/>	Saturday 18:00 No Reboot	Suppress Reboot	Saturday	18:00	Sunday	04:00
<input type="checkbox"/>	Sunday 16:00 Allow Reboot	Allow Reboot	Sunday	16:00	Monday	02:00

Select Automated Patching from the Service Offerings in the Self Service portal.

The Patch Schedule selection is controlled via an automated service request from the Service Desk portal. The use of the Patch Schedule selection also makes it easy to see which machines should have been manually patched or manually rebooted, and then the compliance of those devices can be checked.

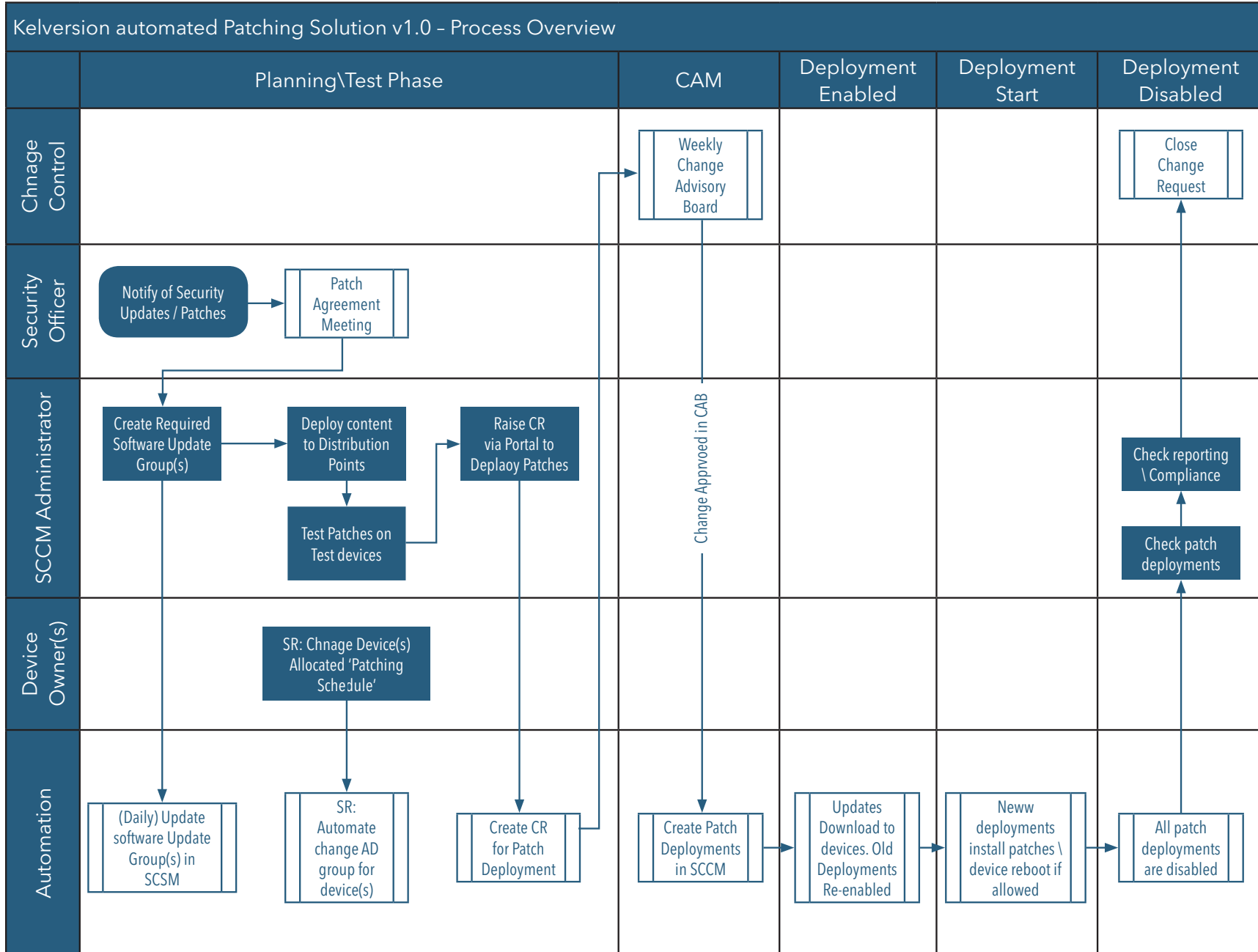
The Solution enables SCCM to raise patch deployment failures as SCOM Alerts, so it is immediately obvious which devices require patch remediation.

REMAINING COMPLIANT

By leveraging the test machines as patch masters, it becomes easy to use the Desired State Configuration functionality in SCCM to determine which devices in an estate are not compliant, and then SCCM can again raise SCOM Alerts to flag the machine to be resolved.

ERRORS HANDLED

Should a patch failure be detected by SCCM, the Solution will automatically create a SCOM Alert. And when combined with Kolverion's Operations Manager 2012 Connectors solution, the SCOM alert will automatically create an Incident Ticket in the Service Desk.



IN CONCLUSION

Patch deployments are essential for the ongoing health of any enterprise estate, but the process can prove disruptive and costly, in both capital and resources, and riddled with procedural gaps. Keverion's Automated Patching Solution offers a reliable, managed approach to patch and security compliance with extensibility to add as many deployment schedules as business needs dictate, while reducing the need for Maintenance Windows and the dedicated attention of the SCCM Admin.

