Apple Lifecycle Management

Introduction

" Ultimately, protecting our customer and employee experience is of utmost importance and worth the investment of a rigorous testing process."

Jennifer Paine
Senior Director Employee Mobility
Southwest Airlines

All types of businesses all over the world are reinventing themselves with mobility. Industries like aviation, law enforcement, and healthcare use Apple devices and apps to carry out essential business tasks. As mobility becomes even more important to the workplace, businesses need to make sure that they're investing in secure platforms and establishing processes to keep those platforms up to date. To sustain this level of stability, businesses are embracing a modern approach to testing and updating software that's proactive, nimble, and perpetual. Moving quickly and continually with software updates creates significant advantages that can keep the enterprise environment a step ahead of emerging security concerns and ensure the highest level of platform integrity. This helps businesses minimize downtime and compatibility issues, better serve customers, and enable employees to have the best user experience.

With macOS, iOS, and iPadOS, it's easy and intuitive for your users to update their devices. With just a tap, they can download and install the latest version of Apple's operating systems. You can also schedule software updates on supervised devices using your mobile device management (MDM) solution.

But before you prompt your users to upgrade, it's critical to test each beta version because your technology ecosystem has unique requirements. And just as important as testing each beta release, key elements in your environment—your IT infrastructure, third-party MDM solution, and business-critical apps—must be ready when a new version is publicly released to your users.

The Apple platform adoption lifecycle includes four ongoing activities for each time a beta is released: preparing your environment, evaluating key areas of Apple's platforms, submitting feedback, and getting ready for your rollout.

When you implement a modern approach to lifecycle management, not only can you be confident when employees update, but you'll also enjoy enhanced data security, improved productivity, maximum uptime, and greater employee satisfaction.

This document is intended to help your IT organization understand the benefits of implementing Apple lifecycle management, consider all the elements that go into managing this process, and establish a clear and repeatable process to ensure that you're ready to stay up to date with the latest software from Apple.

Modern lifecycle management

77%

of iPhone devices introduced in the last 4 years use iOS 13.

● iOS 13   ● iOS 12   ● Earlier Release

As measured by the App Store on January 27, 2020.

79%

of iPad devices introduced in the last 4 years use iPadOS.

● iPadOS   ● iOS 12   ● Earlier Release

As measured by the App Store on January 27, 2020.

Deploying software updates is critical to maintaining the security and integrity of Apple's platforms. Not only does this keep your environment secure, but it allows users to benefit from and enjoy the latest features and security fixes. So it's important for your organization to evaluate all the key areas that work together in your mobile environment, all year long, so you're ready to deploy each release on the first day that it's publicly available.

Southwest Airlines understands the value of taking this approach with its enterprise mobile testing program. Jennifer Paine, Senior Director of Employee Mobility, describes how taking a modern approach is critical:

> "Our pilots and flight attendants depend on iPad devices for critical job functions. We can't risk having an update cause a problem with the iPad. Ultimately, protecting our customer and employee experience is of utmost importance and worth the investment of a rigorous testing process."

Adopting the Apple platform lifecycle in a modern way means embracing the following principles:

Updating software requires an iterative approach

Because of changes in the software landscape, you can no longer delay upgrading your software indefinitely because systems have worked just fine in the past. Instead, organizations are taking a more proactive approach of updating their Apple platforms multiple times a year to stay current.

Apple delivers software updates to keep your devices safe and your existing hardware optimized. macOS, iOS, and iPadOS were developed to make it easy and intuitive for users to set up themselves and update their devices. Enterprises no longer need huge service operations and system imaging processes to make upgrades.

Multiple generations of Apple products can benefit from updates that protect your platform from security vulnerabilities and offer enhanced productivity features.
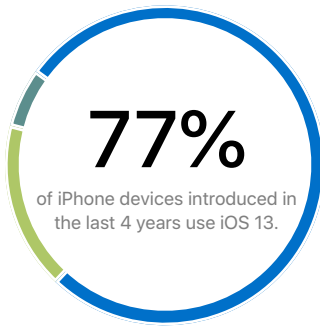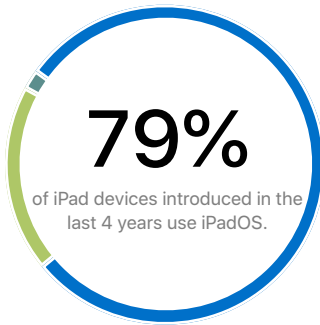
Testing is a year-round effort

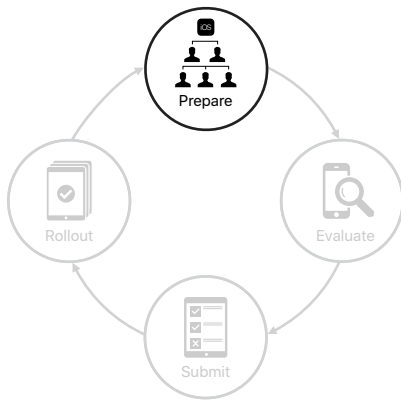Take note of the Apple beta release time frames so you can evaluate each release throughout the year and prepare for any changes that might come up. Whether the release is a major version or a dot release, it's important for users and IT support teams to test and deploy all updates to maximize security and compatibility. Below is an example of iOS release time frames.

12	12.1	12.2	12.2.1	12.3	12.4	12.4.1	13

| June '18 | Sept. '18 | Oct. '18 | Dec. '18 | Jan. '19 | March '19 | May '19 | June '19 |

■ Beta Release   ■ Public Release
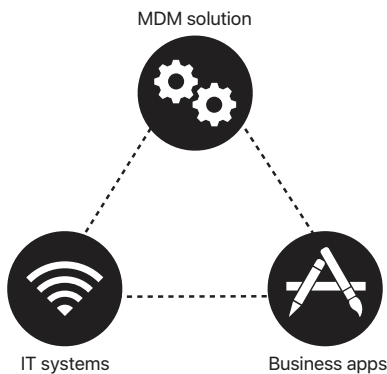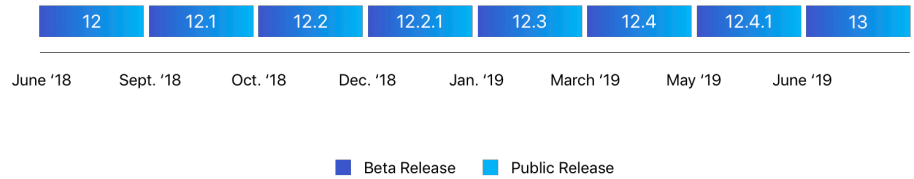
Everything in your mobile environment is interdependent

macOS, iOS, and iPadOS can integrate with just about any IT environment—from network infrastructure to service integration—so you can use Apple devices seamlessly within your environment. Apple software updates can affect the way these services interact with the operating systems and, therefore, impact how Apple devices operate. For that reason, it's important to continue testing upgrades not only on users' devices, but within your larger network ecosystem as a whole.

Adopting a modern lifecycle for Apple's platforms involves the following areas: preparing your environment, evaluating key areas of the Apple platforms, submitting feedback, and getting ready for your rollout.

MDM solution

IT systems

Business apps

Preparing your environment

Establishing a program for managing Apple's platforms involves a few simple steps. First, you'll want to select team members who can be dedicated to your initiative and train them. Second, enroll in the AppleSeed for IT program to get access to prerelease software so you can begin the evaluation process. Finally, establish a device refresh process that includes financing, provisioning devices, and offering the right support.

The Apple platform adoption lifecycle consists of four phases: prepare, evaluate, submit, and rollout.

Assemble teams to support your initiative

Having teams prepared and in place will help you uncover and address potential compatibility issues quickly. Organize a dedicated team focused on evaluating the key areas of Apple's platforms, and form smaller teams of volunteers that represent all your business groups.

Set up a dedicated team

Many businesses have seen the value of adding a dedicated team to their enterprise mobile testing programs to evaluate Apple's platforms. Start by assembling a small group of testers who can devote time to evaluating how macOS, iOS, and iPadOS interact with core business resources, including your MDM solution, Exchange, your network infrastructure, and critical corporate apps. Identify internal talent with software testing or QA experience that could be applied to this role, so they can become part of your existing IT teams. You can also outsource testing to a third party.

Get cross-functional groups to participate

You'll also want to identify employee volunteers who can provide functional expertise in each of your business groups. For example, an airline company may have several business groups, such as ground crew, ticketing, technicians, flight operations, and training. Including each of your business groups helps ensure that you're testing all scenarios—and getting the right feedback—in the field. There's no replacement for using your actual environment to test your devices.

- **Group leaders.** Identify key individuals to serve as group leaders for each functional/business group. Group leaders will gather input and communicate results to your dedicated team. Candidates should have an interest in beta testing and be comfortable managing team members' workflows.

- **Team members**. To build the team, select employees who use Apple devices in their day-to-day roles. The size of each team should be proportional to the overall size of its functional business group.

As you recruit cross-functional leaders and team members, consider:

- How do they use Apple devices and apps in their role?
- Will they have time to participate by performing the same task on multiple devices?
- Will they have time to review documentation, such as release notes?
- Are they capable of installing software, identifying bugs, and submitting useful feedback?
- Are group leader candidates capable of motivating employees and working across multiple groups?

Prepare your employees

After selecting the right employees, use these steps to get everyone up and running quickly:

1. Verify that employees have a personal or Managed Apple ID, know their passwords, and can access beta resources.
2. Ensure that team members are willing to install the beta software on their primary devices—the best way to identify potential issues.
3. Make sure your team is aware that beta testing involves capturing necessary data and submitting feedback as bugs are found.
4. Review best practices for reporting bugs with your team. (See the "Write precise bug reports" section of this document.)
5. Schedule regular reporting and debrief meetings with your group leaders.

Get access to beta resources

Apple offers several ways for your teams to access major and minor releases of Apple beta software and additional resources, including release notes, feedback tools, test plans, product documentation, and program announcements.

AppleSeed for IT program

This beta-testing program, available to any business organization or educational institution, allows you to evaluate the latest prerelease software versions in your unique work environments. Feedback submitted through the AppleSeed for IT program will be funneled to a dedicated review queue. This program also offers detailed test plans and surveys to help you evaluate how new features might work within your environment.

Enroll in AppleSeed for IT with a Managed Apple ID:
appleseed.apple.com

Apple Developer Programs

If you plan to develop, test, and distribute apps on the App Store, sign up for the Apple Developer Program. Custom Apps built internally or developed by a third-party developer and customized for your organization can also be distributed with the Apple Developer Program. Enrolling your organization into this program allows you to build and test your apps so they're ready for release on the new operating system. If your organization has a specific use case that requires private distribution directly to employees through an internal system or a mobile device management solution, apply for the Apple Developer Enterprise Program.

Learn more about the Apple Developer Program:
developer.apple.com/programs

Take stock of your Apple devices

Even the latest versions of Apple's operating systems can run on devices that were released several years ago. So it's important to test the latest beta with all the different models that your organization currently uses. While running a new version on existing hardware is a benefit for companies looking for a good ROI, many organizations are learning that there are also benefits to refreshing devices every couple of years. Regular technology upgrades through leasing reduces compatibility issues and minimizes costs associated with having multiple generations of equipment. You can also trade in eligible devices for credit, then use the funds to lower the cost of new Apple devices or to reduce monthly payments with your carrier. In addition to financing, you'll want to set up your devices using Apple programs and MDM and offer support to your users.

Work with Apple partners and your internal procurement teams who can help manage this lifecycle from end to end for your enterprise. These partners can help with all stages of your device lifecycle, from acquisition of the device through refreshing the hardware. This frees your IT teams to focus on strategic initiatives for the company.
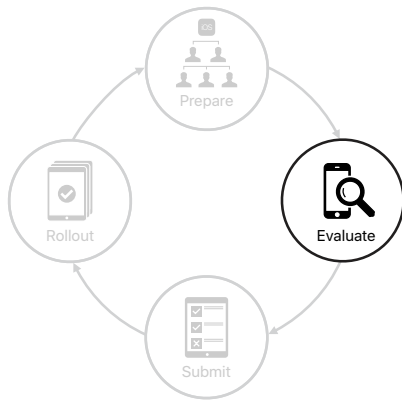
Evaluating Apple's platforms

With all your resources in place, it's time to map a strategy for platform evaluation. Identify the areas of your organization that rely most heavily on Apple's platforms. Determine typical use cases and workflows, and assign those to team members. Then establish a process for evaluating key areas within your organization.

Map your infrastructure

Take inventory of your corporate ecosystem to ensure that all elements will work together smoothly as you test.

IT systems and services

Take stock of all the corporate IT systems and services that need to be evaluated. Validate Exchange functionality, including email, contacts, calendar, tasks, and notes. Test connectivity both inside and outside your network, including Wi-Fi, single sign-on, and VPN, and evaluate Bluetooth device connections and accessories. Validate back-end systems; make sure data servers, middleware, and authentication systems scale efficiently and share their data in a smart way.

Third-party MDM solution

Selecting a third-party MDM solution is essential to managing corporate devices and data. These functions might include but aren't limited to testing configuration payloads, restrictions, and commands with your devices and apps. Understand which types of Apple devices are being used on your network and whether they're owned by the organization or the user so that you can best evaluate your management policies.

App Store and custom apps

App Store and custom apps are central to your company's workflow. Prioritize testing the macOS, iOS and iPadOS apps most critical to your business and users.

- **App Store apps:** Have your dedicated testing team evaluate the basic functionality of all apps, including productivity and collaboration apps as well as built-in essentials such as Mail, Calendar, and Contacts. You may need to work with key developers and use TestFlight to test their apps before they're released publicly on the App Store. Ensure that these apps function properly with MDM and with IT services such as per app VPN and Managed App Configuration. Test new features and make sure that all accessory hardware remains compatible.

- **Custom apps:** Instruct your in-house or third-party developers to plan your testing process. Allow enough time to incorporate new features and to check that apps will work with the new beta once it's been released. TestFlight makes it easy to invite users to test your apps and collect valuable feedback before releasing. Custom apps that are distributed via Apple Business Manager can take advantage of TestFlight. For the smoothest rollout, custom apps should

be validated and ready for users' public operating system releases. Teams should strive to integrate new features within 90 days of a release.

Determine what to test

Now that you've taken inventory of all the key areas within your organization, list all the specific use cases that require testing. In-house or third-party app developers should test custom apps on each new version of beta software.

Many businesses have documented hundreds of use cases to test against each new beta from Apple. Developing this system has helped their teams methodically test new features, possible regressions, and integrations.

Document your use cases

To organize and track ongoing testing, put together a comprehensive spreadsheet of all use cases. Prioritize test cases most critical to your business groups, and outline steps required to test each use case. Assign use cases to team members, who will assign pass/fail comments to each one. Or you can purchase lifecycle management software that coordinates testing and manages requirements, test cases, plans, and bugs.

Here's how you might organize a spreadsheet to track use case testing:

Use Case	Testing Steps	Area	Categories	Group	Employee	Result
Create and send new email	1. Open Mail 2. Tap New Mail icon (bottom right) 3. Enter recipient and subject 4. Tap Send 5. Confirm recipient received email	IT	Exchange ActiveSync	Flight ops, ticketing, training	User 1	Expected
Sync calendar	1. Open Calendar 2. Verify meetings from desktop client show up in Calendar app	IT	Exchange ActiveSync	Flight ops, ticketing, training	User 2	Testing
Push device configurations	1. Exchange ActiveSync payload 2. Managed Mail domains 3. Wi-Fi profiles 4. Certificates 5. Per app VPN profile	MDM	MDM	All groups	User 1	Issue identified
Install a custom app	1. Open company app store 2. Tap the Acme, Inc. app 3. Tap Request button 4. Tap Install when prompted 5. Confirm app has been installed	Apps	Custom apps, MDM	All groups	User 3	Expected
Open file stored in document provider app	1. Open document provider app 2. Tap a folder to open it 3. Tap a file inside 4. Confirm file opens	Apps	App Store apps, MDM	Flight ops, ticketing, training	User 3	Testing

Review your custom apps

If your company develops its own custom apps, make sure your in-house or third-party development teams evaluate these apps with each beta release to determine the impact of revised APIs and programming languages. To perform this verification, Apple offers early access to beta versions of Swift, Xcode, and operating systems through Apple Developer Programs. Consider the following guidelines as part of your evaluation process.

- **Stay up to date on announcements.** Get the latest development updates, tips, and how-to information by visiting developer.apple.com/news/. View WWDC session videos at developer.apple.com/videos/ to learn more.

- **Review release notes.** Download and review release notes for each new beta release, and distribute a summary to team members for greatest impact.

- **Check whether APIs, Swift, or Xcode have evolved.** The earlier you identify potential changes, such as deprecation or changes in API semantics, the more time you have to implement and test these changes.

- **Test for compatibility.** Test apps against the unique software configurations your users may have, such as the previous version, the latest version, and the current beta. Test devices with the same architecture, screen resolution, and PPI for each major OS version your app aims to support.

- **Conduct a visual test.** See how your apps perform in terms of resolution, pixelation, layout, alignment, and orientation.

- **File bug reports.** While discussing issues on the Apple Developer Forum is helpful for exchanging information, it's not a substitute for a bug report. Filing a bug report helps ensure that an actual framework bug can be diagnosed and fixed before an Apple update goes public. (See the "Submitting feedback" section for more details.)

- **Implement new standards.** When new standards are announced, observe the required implementation date for each. This is especially important if you intend to submit apps to the App Store.

Learn more about testing your apps on beta OS releases:
developer.apple.com/library

Establish your evaluation process

Set up a unified process for evaluating key areas within your organization so all teams are aligned.

Receive notifications of a new beta build

Don't be afraid of the first beta or wait for a later version. Test and give feedback early. The sooner Apple receives your feedback, the sooner changes can be made. Stay on top of new operating system beta releases by subscribing to the RSS feed from developer.apple.com/news. Though release schedules vary, familiarizing yourself with prior years' beta release timelines can help you forecast your team's upcoming needs for availability.

Review the scope of changes

Before you install any beta software, it's critical to review release notes for fixes and new features to determine which changes might affect your Apple ecosystem. Check Apple's website and the AppleSeed for IT portal for announcements or documentation on updates, and share prerelease information with your teams to ensure that all members are informed of changes.

Prioritize user groups, use cases, and critical apps

Determine which user groups have business-critical functions. Prioritize testing for those groups and focus on evaluating the use cases and apps that have the biggest impact on those teams.

Balance team priorities

Each week, determine the availability of your dedicated team members as well as cross-functional group leaders and their teams. Have your dedicated team start each testing procedure before engaging group leaders and team members from business-critical groups.

Submitting feedback

By providing feedback to Apple engineering and AppleCare, you allow Apple to identify issues impacting your unique environment, fix issues critical to your organization, and make macOS, iOS, and iPadOS even better for your users.

Submit feedback to Apple

Apple offers a robust tool for you to submit your feedback. Feedback Assistant is available on iPhone, iPad, Mac, and the web, making it easier to submit effective bug reports and request enhancements to APIs and tools. Filing feedback as early as possible gives you the best chance at having your concerns addressed in the public release. The app lets you capture events right at the point of failure. When you log in to the app and submit your feedback, you'll receive a Feedback ID, which you can use to follow up with your team and Apple.

Feedback Assistant app

This app lets you capture events right at the point of failure and report anything that isn't functioning properly with your IT systems and services or MDM policies. For example, use the Feedback Assistant app if your custom app isn't tunneling properly with VPN. On an iOS or iPadOS device, launch Feedback Assistant from the Home screen, then capture events right at the point of failure. On a macOS computer, launch Feedback Assistant from Finder. When you sign in to the app and submit your feedback, you'll receive a Feedback ID, which you can use to follow up with your team and Apple. You can also view previously provided feedback under the Submitted section in the app.

Feedback Assistant website

Use this tool to submit any bugs related to your custom app, request enhancements to APIs and developer tools, and track your submissions. Problems with the behavior of your custom apps can result from a framework bug in the beta or a framework change that exposes a bug in your app's code. If the issue looks like a framework bug, your in-house or third-party app development team should attach sample code you created (as a runnable Xcode project) to a bug report, then submit using the Feedback Assistant for Developers website.

Learn more about Feedback Assistant:
feedbackassistant.apple.com

Note: Bugs logged through Feedback Assistant associated with a personal Apple ID won't end up in the prioritized queue for AppleSeed for IT. If Apple devices or apps aren't functioning properly on your IT systems and services or MDM policies, use the Feedback Assistant app with a Managed Apple ID associated to your organization.

Write precise bug reports

Submitting bug reports as early as possible in the beta process is the most effective way to get your issues fixed. Be as specific as possible when submitting bug reports and limit each submission to one issue. Providing as many details as possible, such as those listed in the suggestions below, allows Apple to effectively distribute information to specific engineering teams and avoid requests for additional information. If more information is required, someone from Apple will reach out to you.

- Give your bug a descriptive title so that Apple teams can easily reference your issue.

- State clearly what you expected to happen, what happened instead, and why you think it's a problem.

- Submit screenshots and/or a video showing on-device behaviors and UI examples.

- List clear and concise steps needed for Apple to reproduce the issue.

- Collect and attach any iOS and iPadOS device logs or macOS console logs relevant to the issue.

- Provide a reproducible test case, if possible.

Learn more about logs, reproducible test cases, and other details: appleseed.apple.com/sp/help/feedback

Review your feedback

Use the Feedback Assistant app to view feedback you've filed or saved as a draft. You can also see if any feedback requires additional action or information on your part and receive notification that an issue has been fixed in a new beta release. Review AppleSeed for IT release notes for each beta to see highlights of issues fixed since previous betas.

Get assistance from AppleCare

With AppleCare for Enterprise or an AppleCare OS Support agreement, you can verify that issues from previous software releases are fixed in a new beta. You can request testing assistance from the AppleCare Enterprise Support team or an Apple Systems Engineer (SE) and notify them of deployment-blocking issues you report through the Feedback Assistant app. These experienced Apple agents can quickly guide you through testing, submitting feedback, and tracking any issues.

Note: Feedback logged through AppleCare will not end up in the prioritized queue for AppleSeed for IT. It's recommended that you also submit feedback through the Feedback Assistant app with a Managed Apple ID associated with your organization.

Work with your solution providers

To ensure that your apps and devices work properly with beta OS releases, it's crucial to engage third-party vendors that support Apple's platforms and test their beta solutions in parallel.

MDM solution providers

Make sure your MDM solution provider's platform will continue to support new versions of macOS, iOS, and iPadOS betas and follow their timetable for new feature support. If your MDM vendor offers a beta version of their solution, participate in their beta program as well. Ideally, you should be testing the management capabilities of your MDM solution in beta on devices that are also running the latest beta release. That way you get a full view of what the experience will be for employees.

App developers

Because your employees rely on apps to accomplish their day-to-day roles, notify app developers of compatibility issues between key apps from the App Store and the new beta. This also applies to any other software solutions used by your organization.

Other vendors

Ask vendors that supply your networking, VPN, Bluetooth device connections, and accessories to ensure that your Apple devices work with your organization. Your network provider can help you determine an evaluation and rollout strategy when you're preparing for a network upgrade.

Getting ready for your rollout

Prepare

Rollout

Evaluate

Submit

Once the new macOS, iOS, and iPadOS version has been publicly released by Apple, test it, encourage users to install it once it's certified, and educate employees on new features.

Certify the public release

When a new software version is publicly released, move quickly to evaluate it. Once you're confident that things are running the way they should be, encourage users to update their Apple devices as soon as possible.
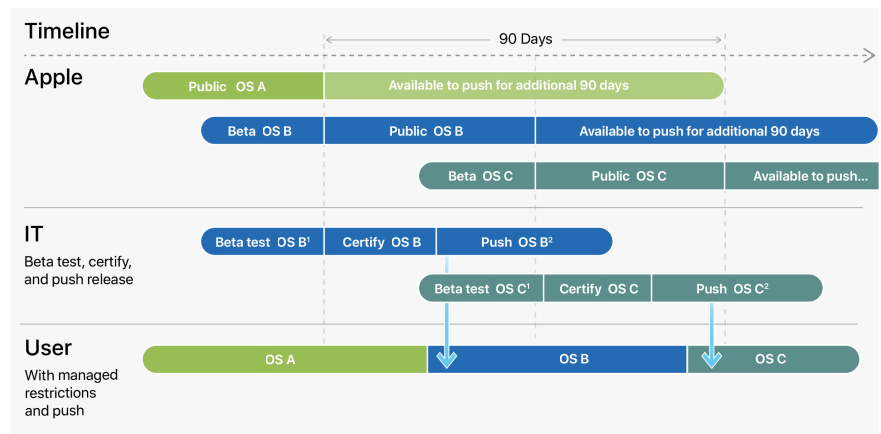
Involve your entire team

Even though your teams have been consistently testing each beta release, it's essential to conduct a thorough final evaluation of the public release before users update their devices. Your dedicated team should prioritize evaluation of the latest release, methodically testing all critical use cases. In parallel, have team members from participating business groups test critical use cases as well.

Schedule software updates

Though you want your employees to update their devices to the latest version as soon as possible, there may be instances in which a new version is released while you're still certifying a previous version. With the latest versions of macOS, iOS, iPadOS, and even tvOS, you can prevent users from updating their devices over-the-air to the latest operating system for a short period so you have the time and flexibility you need to complete a thorough certification. Then once you're ready, you can choose which version you want users to download and install, and you can push it to them directly.

In this scenario, IT has previously tested, certified, and pushed version "OS A." The following illustrates how you can manage additional releases when an MDM restriction is used on supervised devices.

Timeline		90 Days		
Apple	Public OS A	Available to push for additional 90 days		
	Beta OS B	Public OS B	Available to push for additional 90 days	
		Beta OS C	Public OS C	Available to push…
IT Beta test, certify, and push release	Beta test OS B[1]	Certify OS B	Push OS B[2]	
		Beta test OS C[1]	Certify OS C	Push OS C[2]
User With managed restrictions and push	OS A	OS B		OS C

1. During the Beta testing phase, you can place an MDM restriction for a specified time that prevents users from manually updating their device once a version is publicly available.

2. You don't have to wait until the restriction expires to push out a software update once you've certified it. However, at the end of the delay period, users will get a notification to update to the earliest version that was available when the delay was triggered.

These features will allow you to restrict and push updates according to your organization's needs:

Managed software updates

For any supervised macOS, iOS, iPadOS, and tvOS devices, you can send an MDM restriction that prevents users from manually updating a device over-the-air for a specified time. Users can still update their devices with Apple Configurator or Finder on macOS if they've been granted access for iOS, iPadOS, and tvOS devices. When you implement this restriction, the default delay is 30 days, and is triggered the moment Apple releases an operating system update. However, you can change the default number of days you prevent updates, anywhere from one to 90 days. When the delay expires, users get a notification to update to the earliest version of the operating system that was available when the delay was triggered.

Initiated software updates

Once you've certified a software version, it's recommended that you use an MDM command to push a software update to supervised devices, prompting users to update their devices. Your MDM solution will now provide a list of software updates available for you to push. If you push a software update while users are under the managed software update restriction, the restriction becomes active again the moment the next software update is publicly available from Apple. You can also use this MDM command to separate the download and installation of updates to avoid disrupting use of the device and allow users to install at a more convenient time. Installing at a different time also allows you to notify your employees before you push an update.

Note: On macOS devices, the software updates are limited to operating system updates only. Safari, security updates, or other supplemental updates for macOS aren't restricted.

Learn more about how to best Update your iPhone, iPad, or iPod touch: support.apple.com/HT204204

Communicate next steps with employees

Here are some considerations for communicating with users about updating their Apple devices and installing key apps. Remember to be transparent when communicating with your entire organization, including with your in-house or third-party app developers.

- On launch day, send an email, update your web page, or send the announcement using an internal messaging tool. You might want to note areas that still may not be working as expected.

- Publish FAQs and support information to your internal website or wiki, including fixed issues, knowledge-base articles, information about new features, and the best way to report issues.

- Schedule follow-up brown bag sessions, launch calls, and webinars.

- Promote the importance and value of continued beta testing to all functional organizations, and share the beta information with all who need it.

Educate users on the new features

If your users understand how they can benefit from and enjoy new macOS, iOS, and iPadOS features, they're much more likely to upgrade quickly. Let them know that the new release helps keep their devices safer and helps them be more productive. In your communications, be sure to include links to internal and external resources that provide additional information about the latest tools, features, and apps from Apple.

Summary

Apple prides itself on delivering the best technology to the end users. You can put this power to work for your enterprise through proper management of the Apple platform lifecycle. Methodical testing of your apps and ecosystem throughout beta releases allows you to take advantage of the public releases as early as possible, providing new features, enhanced security, employee productivity, and operational integrity.

Learn about macOS deployment, management, and security features:
support.apple.com/guide/deployment-reference-macos

Learn about iOS and iPadOS deployment, management, and security features:
support.apple.com/guide/deployment-reference-ios

Learn about mobile device management settings for IT:
support.apple.com/guide/mdm

Learn about Apple Business Manager:
support.apple.com/guide/apple-business-manager

Learn about Managed Apple IDs for Business:
apple.com/business/docs/site/
Overview_of_Managed_Apple_IDs_for_Business.pdf

Learn about Apple at Work:
www.apple.com/business/

Learn about IT features:
www.apple.com/business/it/

Learn about Apple Platform Security:
www.apple.com/security/

Browse available AppleCare programs:
www.apple.com/support/professional/

Discover Apple Training and Certification:
training.apple.com

Engage with Apple Professional Services:
consultingservices@apple.com