iOS Lifecycle Management

A Modern Approach to Platform Readiness

Contents

Executive Summary

"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 are using iOS 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 business minimize downtime and compatibility issues, better serve customers, and enable employees to have the best user experience.

With iOS, 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 iOS. You can also schedule software updates on supervised devices using your mobile device management (MDM) solution.

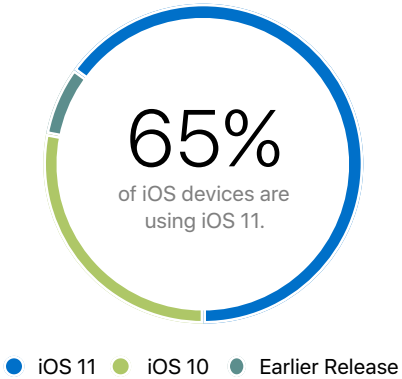
But before you prompt your users to upgrade, it's critical to test each beta version of iOS because your technology ecosystem has unique requirements. And just as important as testing each beta release, key elements in your environment—your IT infrastructure, third-party MDM solution, and business-critical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

The iOS platform adoption lifecycle includes four ongoing activities for each time a beta is released: preparing your environment, evaluating key areas of the iOS platform, 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 iOS 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.

A Modern Approach to Lifecycle Management



As measured by the App Store on January 18, 2018

Deploying software updates is critical to maintaining the security and integrity of the iOS platform. Not only does this keep your environment secure, but it allows iOS 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 their 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 cannot 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 iOS platform lifecycle in a modern way means embracing the following principles:

Updating software requires an iterative approach

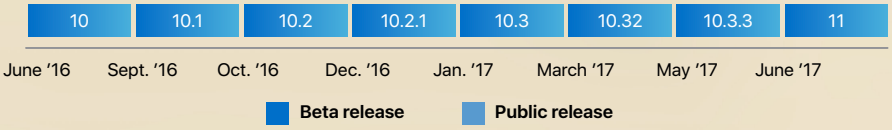
Because of changes in the software landscape, you can no longer delay upgrading your software indefinitely because systems have worked just fine in the past. Instead, organizations are taking a more proactive approach of updating the iOS platform multiple times a year to stay current.

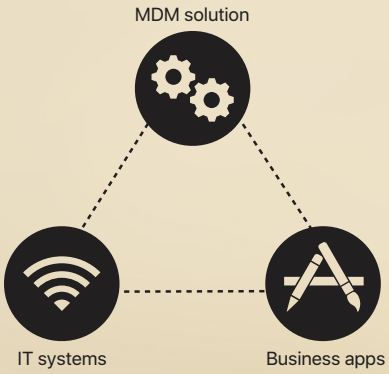
Apple delivers software updates to both keep your devices safe and your existing hardware optimized. iOS was developed to make it easy and intuitive for users to set up themselves and update their devices—enterprises no longer need huge service operations and system imaging process to make upgrades.

Multiple generations of Apple products can benefit from iOS updates that protect your platform from security vulnerabilities and offer enhanced productivity features.

Testing the iOS platform is a year-round effort

Take note of the iOS beta release timeframes so you can evaluate each release throughout the year and prepare for any changes that might come up. Whether the release is a major iOS 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 timeframes.

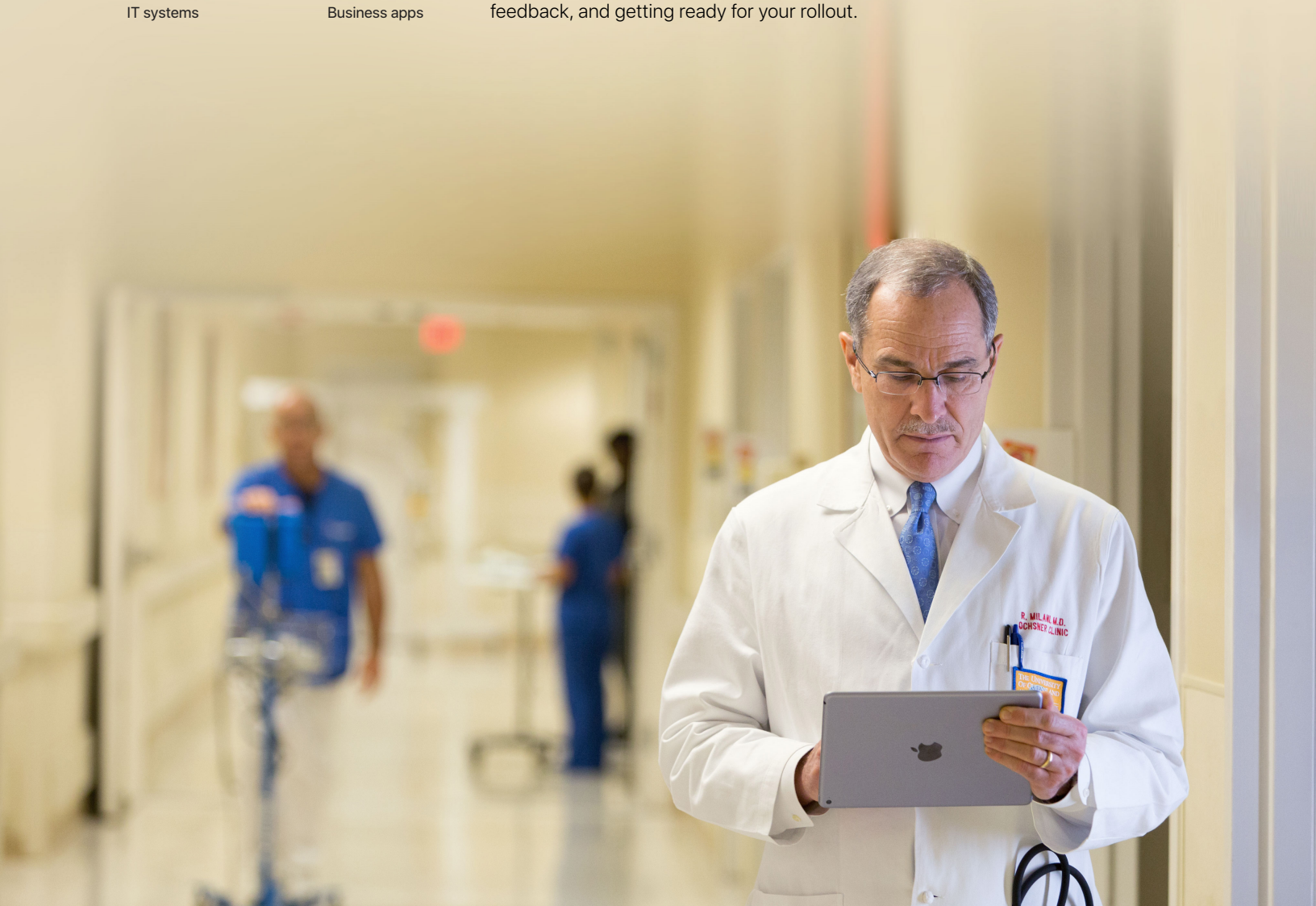




Everything in your mobile environment is interdependent

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

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



Preparing Your Environment

The iOS platform adoption lifecycle consists of four phases: prepare, evaluate, submit, rollout

Establishing a program for managing the iOS platform 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 program to get access to prerelease iOS software so you can begin the evaluation process. Finally, establish a device refresh process that includes financing, provisioning devices, and offering the right support.

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 the iOS platform, 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 the iOS platform. Start by assembling a small group of testers who can devote time to evaluating how iOS interacts with core business resources, including your MDM solution, Exchange ActiveSync, 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 mobility and/or 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 possible 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 out 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 these criteria:

- 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 an 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 iOS 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. Your Apple Systems Engineer and AppleCare Account Manager can help enroll you in AppleSeed for IT.

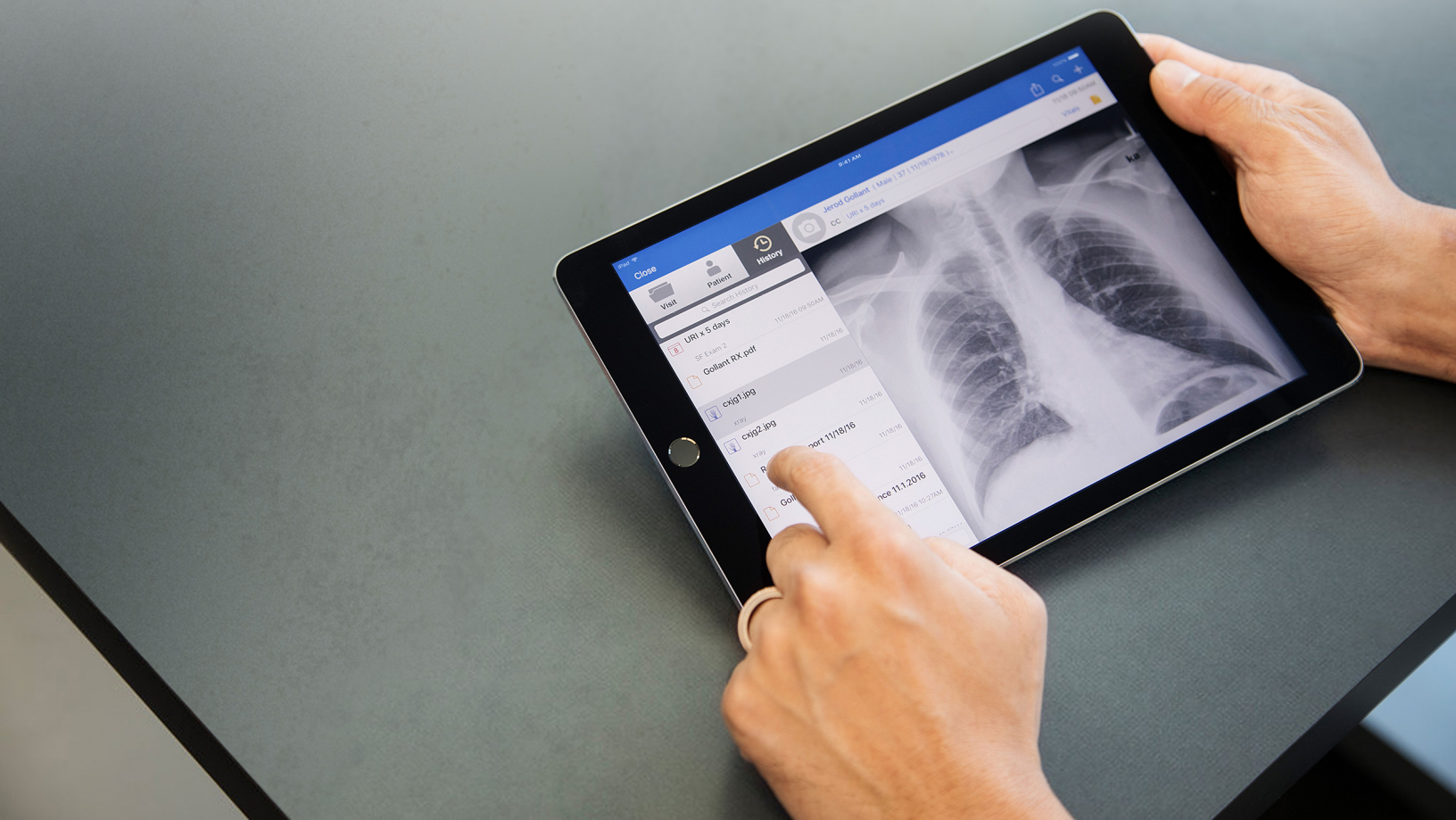
Apple Developer Programs. If you plan to develop, test, and distribute iOS apps on the App Store, sign up for the Apple Developer Program. If your custom apps will be distributed within your enterprise, enroll in the Apple Developer Enterprise Program instead. Enrolling your organization into this program allows you to build and test your apps so they're ready for release on the new OS.

Take stock of your iOS devices

Even the latest versions of iOS 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 is currently using. While running a new iOS 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 two or three 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 smartphones 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 up your IT teams to focus on strategic initiatives for the company.

- Learn more about Apple Financing.
- Learn more about Apple Recycling.
- See iOS Deployment Overview for Business and iOS Deployment Reference for more detailed information.
- Learn more about AppleCare programs.



Evaluating the iOS Platform

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

Map out 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 ActiveSync functions, 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 iOS 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. iOS apps are central to your company's workflow. Prioritize testing the App Store apps and custom 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 iOS accessory hardware remains compatible.
- **Custom apps.** Instruct your in-house or third-party developers to plan out 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. Have your teams use an ad hoc provisioning profile to export an Apple Developer Enterprise Program app from Xcode for beta testing. After exporting any beta apps, they should consider using Xcode Server to distribute them to testers and other team members. For the smoothest rollout, custom apps should be validated and ready for users' public iOS release. 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 the iOS beta.

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 ecosystem integration.

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 shows 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 insider 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 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 iOS 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 https://developer.apple.com/news/. View WWDC session videos at https://developer.apple.com/videos/ to learn more.
- **Review release notes.** Download and review release notes for each new Apple 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 iOS version, the latest iOS version, and the current iOS beta. Test devices with the same architecture, screen resolution, and PPI for each major iOS 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 the iOS 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.

For more detailed information, review the technical note, Testing your apps on Beta OS releases.

Establish your evaluation process

Set up a unified process for evaluating key areas within your organization so all teams are on the same page.

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 iOS 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 iOS 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 iOS even better for your users.

Submit feedback to Apple

You'll want to select the right Apple tool to file your feedback depending on the issue you're encountering. Write clear and concise feedback reports, and provide additional information needed by Apple's engineering team.

Select the right feedback tool. Apple offers several tools for you to submit your feedback; choosing the right one expedites delivery to the engineering teams. Filing feedback as early as possible gives you the best chance at having your concerns addressed in the iOS public release.

- **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 device, launch Feedback Assistant from the Home screen, then 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. You can also view previously provided feedback under the Submitted section in the app.

- **Bug Reporter tool.** Use this tool to submit any bugs related to your custom app, request enhancements to iOS 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 Apple's Bug Reporter tool.

Note: Bugs logged through bugreporter.apple.com won't end up in the prioritized queue for AppleSeed for IT. If iOS devices or apps aren't functioning properly on your IT systems and services or MDM policies, use the Feedback Assistant app.

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 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 for iOS at https://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 the iOS 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.

Work with your solution providers

To ensure that your apps and devices work properly with beta iOS releases, it's crucial to engage third-party vendors that support your iOS platform 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 the iOS beta and follow their timetable for new iOS 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 out 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.

235K

There are over 235,000 business apps on the App Store.

App developers. Because your employees rely on apps to accomplish their day-to-day roles, notify app developers of compatibility issues between key iOS 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 iOS devices are working 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

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

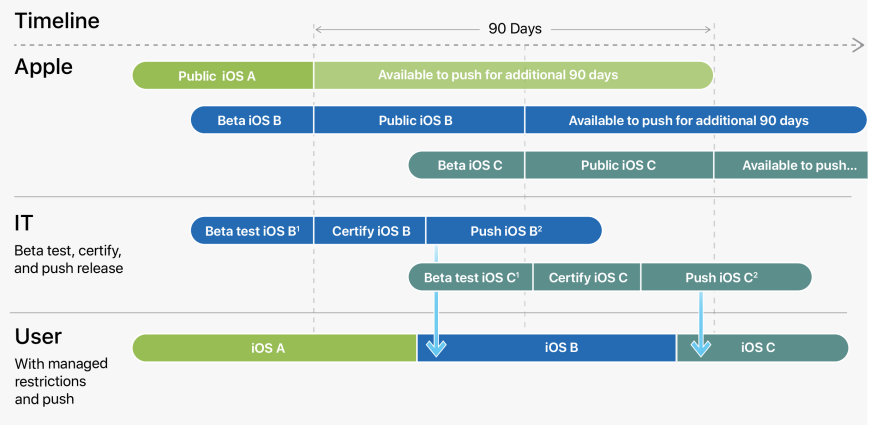
Certify the public release

When a new version of iOS 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 iOS 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 latest version of iOS as soon as possible, there may be instances in which a new version of iOS is released while you're still certifying a previous version. With iOS 11.3 and later, you can prevent users from updating their devices over-the-air to the latest OS for a short period so you have the time and flexibility you need complete a thorough certification. Then once you're ready, you can choose which iOS update 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 out version "iOS A." The following illustrates how you can manage additional releases when an MDM restriction is used on supervised devices.



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 of iOS 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 iOS device, 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 iTunes if they've been granted access. When you implement this restriction, the default delay is 30 days, and is triggered the moment Apple releases an iOS 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 iOS that was available when the delay was triggered.

- **Initiated software updates.** Once you've certified a version of iOS, it's recommended that you use an MDM command to push out 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 out. If you push out 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 out an update.

Learn more about how to best Update the iOS on your iPhone, iPad, or iPod touch.

Communicate next steps with employees

Here are some considerations for getting the message out to users about updating their iOS 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/or 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 know about how they can benefit from and enjoy new iOS 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. The following resources can help your users get the most out of their iOS devices:

- Learn more about iOS.
- View user guides for iPhone and iPad.
- Learn tips and tricks on iOS.
- View tips for business apps on iTunes.
- Explore business apps on iTunes.
- Discover business apps in Apps in Business Getting Started Guide.
- Download the Apple Support app.

Summary

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

TM and © 2018 Apple Inc. All rights reserved. Apple, the Apple logo, iPad, iPhone, iTunes, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. AppleCare and App Store are service marks of Apple Inc., registered in the U.S. and other countries. Other product and company names mentioned herein may be trademarks of their respective companies.