RB Consulting, Inc.

IOS App Store Publishing Overview

Mobile Applications

Prepared by:
Robert Broadhead

April 15, 2015

RB Consulting Confidential

Revised: Rob Broadhead

No portion of this document may be reproduced, stored in a retrieval system or transmitted in any form by any means without the prior written approval of RB Consulting Inc. Any such requests should be sent to:

RB Consulting, Inc.
309 Inwood Way
Franklin, TN 37064

Contact Name: Rob Broadhead

In no event shall RB Consulting be liable to anyone for special, incidental, collateral, or consequential damages arising out of the use of this information.

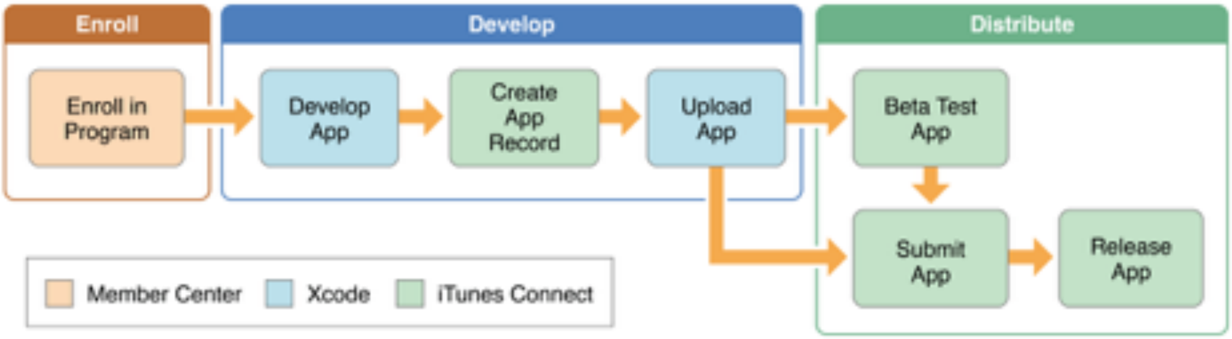
This document contains RB Consulting sensitive material. Posting or sharing this material outside of RB Consulting should be done only at management discretion.

Printed in the United States

iOS Overview

Apple provides the App Store for purchasing both PC and mobile applications. The ins and outs of publishing is somewhat similar for all application types, but this document is going to address only mobile development. Currently mobile devices come in a few varieties: iPhones (roughly 3 supported types), iPad, iPad Air, iPod touch and iPad mini. The devices all have a normal and retina (high resolution) screen model. The key differences in these devices are screen size and resolution.

Apple requires all publishers to at least have an iTunes account and a developer subscription. The developer subscription is $99 per year and can cover safari development, Mac apps, or mobile apps. The flavors of developer subscriptions limit the tools and publishing targets available so a mobile development account is needed for developing and publishing to mobile devices. The application process for an initial development account is not instantaneous and requires the user to verify they are a real person or business. The process provides an easy to use interface that walks through the application steps, but it does take a few days of elapsed time to get verified and ready to publish. It is important to get all the long term information settled on early in the process as changing publisher data (eg. company name change) is very difficult. DBA names are not allowed so a company cannot publish under multiple names unless those are all companies in their own right.



Once a developer account has been created, there are several configuration options that need to be setup including: developer access; mobile devices that are available for testing; and some other minor options. There is adequate documentation for these steps from Apple so the details are not covered here, but mentioned as a warning that it is not an instantaneous process. The full details can be found at: https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/ManagingAccounts/ManagingAccounts.html#//apple_ref/doc/uid/TP40012582-CH24 (Managing Accounts).

Development Requirements

Development for any Apple devices has a large amount of documentation available and a well defined SDLC process. There is a also a sizable and active development community that can be tapped for assistance in learning as well as trouble shooting.

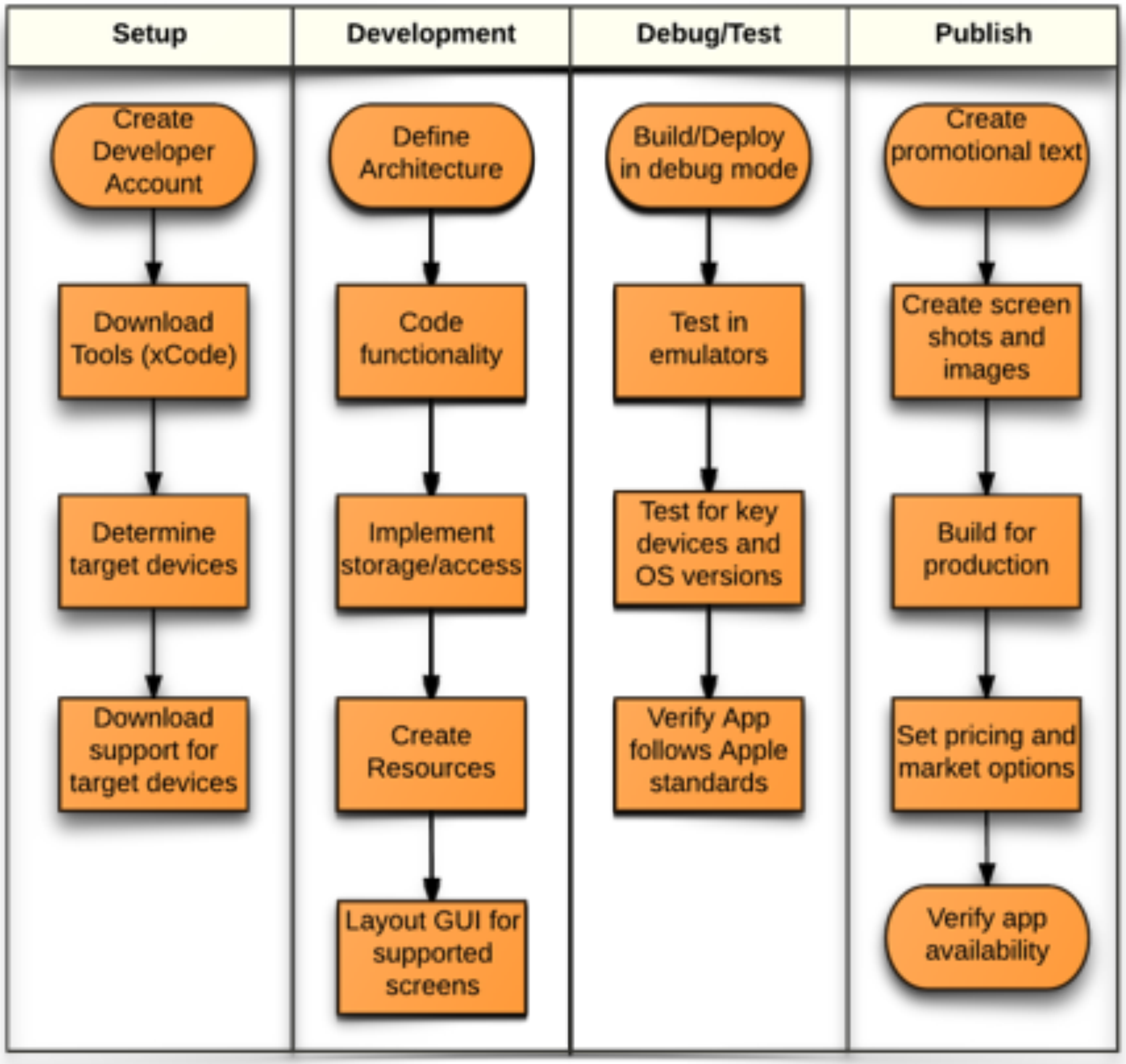
Apple tries to enforce quality from all of their developers and provides tools to assist them deliver high quality applications. Although there are third party tools available the easiest way to conform to "the Apple way" is to use the tools they provide.

The Apple tools unfortunately are all Mac OS based so alternatives will be required if development will only be done on another OS. Since those are not Apple supported methods it is important that due diligence is done to ensure any tools selected have a known record of success in developing for iOS.

Once the tools are selected and installed there are a typical set of steps to follow to produce and publish and application. At the highest level these steps are:

- Setup/Install the development environment
- Develop the application
- Test/Debug
- Publish

Each of these steps has a number of sub steps noted in the image below:



There are a number of ways to complete the first three categories of steps, but the final category (Publish) requires the Apple described approach to be followed.

Publishing Steps

Apple provides a series of screens to gather information during the publishing process. The screens tend to be clear on what is needed. The key items are listed here to help prepare for the publishing process. It is important to note that Apple reviews all published applications and updates so it will save time and frustration to ensure all content meets their guidelines. App store content and promotional text can be updated in almost real time, but fix releases and patches can take weeks to be updated in the store. Here are the key items needed to publish an application:

- Determine content rating and country (or countries) of distribution
- Verify platform and screen compatibility targets
- Price model (amount-tier, in app billing)
- Promotional graphics, screen shots, and content
- Prepare for Release (gather App Store content)
- Build and Deploy (version information)
- Complete App store listing and verify display
- Final checks and send for review
- Apple will either accept the app and make it available or reject it and provide information on why it failed

Publishing Details

The publishing process requires a number of questions to be answered about the application to be published. These answers effect how the application page will look on the app store, how it will appear in searches, and what sort of searches will list the application as a result. Proper wording and design of this content is a key part of making the application available to the target market and appealing to potential users. It can be considered the "Super Bowl Ad" for the application being published.

There is no such thing as an application that is "sort of" on the App Store so even applications that are intended for a small, specific audience still have to meet all of the requirements in order to be available. Apple lists limited user availability via geography or other reasons as a potential reason for rejection so the app store is not going to be a valid delivery method for "internal" applications. There are other ways to distribute applications if the application is, for example, only intended for internal use or otherwise intended to be unavailable to the general public. Apple provides testing options to deliver an application to select users before general release, but this is a temporary state and not allowed to be treated as the final delivery mechanism.

Preparing for Release

The development process for mobile is recommended to be done in the Mac environment (as noted when discussing tools). Whether using Apple's tools or ones from another vendor the development process is typically done heavily within the tool and details can be found in the provided tool documentation. Apple has a large number of examples and tutorials available for their tools. Third party tools may have great documentation as well so utilize that where possible. There are five main tasks involved in publishing :

1. Gather materials for release
2. Prepare for Upload
3. Archive and Validate the app.
4. Prepare remote servers
5. Upload the app.

Gather Materials

Gathering materials for release is not much of a development effort (if any) and is really a sales/marketing/product placement exercise. During the gather materials phase the following information/resources need to be created/gathered (or decisions made):

- Pricing Tier/Discounts
- In-App purchases
- Name and screen shots (4.7”, 5.5”, 4”, 3.5”, and iPad versions), video previews if desired
- Description (up to 4000 characters)
- Keywords (up to 100 characters)
- Support, Privacy and Marketing URLs
- App Icon and copyright info
- App categories
- Reviewer information (contact and review account)

Apple has a full set of documentation recommending an approach to these items and a number of restrictions as well. It will help to review that documentation to avoid issues during the application review process. The notes about the textual content and screen shots technical details should be noted to help avoid application rejection.

Preparing for Upload

There are a few steps required in order to upload an application before starting the upload process. Apple has a full set of documentation recommending an approach to these steps and a number of restrictions as well. It will help to review that documentation to avoid issues during the upload process. If development has been done in Xcode it is an easy process to perform and well documented. Uploading is a step that cannot be reversed so be sure to validate the following before proceeding:

- Interface and Store guidelines
- Itunes Connect information (content gathered in prior step)
- Project information (config and links set for production)

Archive and Validate the App

The application is uploaded in a signed archive format. This signed archive is easily built from Xcode. The archive's purpose is to provide a duplicate of the production version for debugging and to allow the application to be tested as a production archive before submission.

The validation is done first by the archive creator to verify it is a valid archive and then by iTunes connect. The iTunes connect validation produces a distribution certificate and provisioning profile if the archive is properly signed and the expected format. This process helps speed review time for Apple as it is a sort of "pre-upload" that gets reviewed by an automated system. All errors raised during validation must be corrected before the certificate and provisioning profile are created.

Prepare remote servers

Apple provides a lot of tools to help ensure a published application works as close to 100% as possible. One main facet that they do not validate is the production servers and links an application might use. This is also a facet that is not always immediately apparent to a user. As part of the upload process ensure that all configuration options, links and connections are pointed to the correct servers before the final upload. A test account is required for the review process so ensure that the account is enabled and fully functional in the production environment.

Upload the Application

Once all of the build and validation steps are completed, the application can be uploaded to the Apple Store for review. There is a review required and completion times vary by time of year and size/complexity of the application. Apple states also that more expensive applications get a more thorough review, but that is not to say free applications will sail through the review process. It is not clear what all of the steps behind the scenes are, but it appears that there are human reviewers involved in the review process (it is not strictly automated) and they will use provided test account data to verify functionality. Check the Apple Developer site for information on time frames and quiet periods.

Coding Standards

Apple strongly guards the "Apple experience" and has a list of features that they try to incorporate into all of the third party applications to protect the Apple brand and experience. These requirements have a lot more to do with improved user experience than easing developer effort so be prepared to have to err on the side of quality when putting together an iOS application.

One of the benefits of targeting iOS devices is that Apple tends to limit options available to consumers and the larger devices have built in support for apps aimed at smaller screen sizes (phones). The version of the operating system tends to be pretty stable and users are practically forced to upgrade as new versions come out so there is rarely more than a version or two of the operating system that will have to be supported for a release. There is still a need to decide what devices will be supported as that will effect the store options chosen for display and testing considerations.

User Interface

The UI design standards are intended to keep a consistent look-and-feel for Apple iOS applications. The design guidelines are rarely hard and fast considerations, but tend to be more about aesthetics (art rather than science). In general here is a list of what Apple is looking for :

1. Use controls designed for a touch interface
2. Create layouts that work across the various screen sizes
3. Hit targets should be at least 44x44 points in size
4. Text sizes should be at least 11 pt for ease of viewing
5. Ensure contrast and spacing make for ease of reading text
6. Provide high resolution versions of images so they will look good on a retina display
7. Keep size ratios in mind to avoid image distortion
8. Use easy to read control layouts with controls close to the item(s) they modify

Behavior and Permissions

Mobile devices have a number of hardware features that an application may need to access. The list items below are not hard and fast rules about access, but the list of do's and don'ts for These features include: camera, speaker, ports (USB or other), display and other options. Developing a quality application leads to these considerations :

1. Do not use a resource unless it is required for the application
2. Do not access user data without specific permission granted
3. Limit resource access where possible, in particular be mindful of and reduce network traffic where possible
4. Access to resources should be limited to when the application is active unless background usage is a part of the application
5. The user should always be made aware of what the application is doing via a clear interface and messages
6. Do not use a resource longer than needed
7. Enforce "clean-up" strategies so a crash does not lock access to a resource

Performance and Stability

The standards for performance and stability are as high as any other facet of an application. In most cases failure to meet the standards will result in rejection of an application. The notable stability requirements are:

- App should not crash
- No bugs exhibited
- App must perform as advertised
- Undocumented or hidden features are not allowed
- Non public APIs are not allowed
- Apps cannot download code
- Apps that are primarily advertising or promotional will not be allowed
- Apps cannot be greater than 100MB
- Apple services must be used where applicable (webkit, iOS Data storage)**Testing Notes**

Apple requires registration of devices that are used for testing. This can be a little confusing to new developers and non technical users that will be used to test the application. The developer is required to have the UDID of all test devices entered into the provisioning profile before a build (devices can be added later but a new build is required when a device is added). This is found by connecting the device to iTunes and clicking on the Serial Number field in the device summary. Clicking will scroll the display through a few id values including UDID. The UDID is long and not intuitive so a copy-paste step is the best way to get the value to the developer. The developer will then need to enter the UDID value into the provisioning profile on iTunes connect.

A generic test script is beyond the scope of this document, but there are some key points that should be hit during testing an application. These steps include:

1. Navigate to all parts/screens of the application
2. Verify rotation to and from landscape/portrait works without impacting app.
3. Verify all back buttons
4. Verify pause/sleep and restore functionality works for apps (background app and power off/on)
5. Verify all resources (images, etc) work properly on publish
6. Verify iPhone targeted apps work in 2x mode on a tablet (This should be done even if an iPad version exists)
7. Test the application interaction with voice over features where applicable
8. Test the application integration with voice enabled control such as Siri
9. Test the application in high usage states such as many applications running at the same time.
10. For resource usage such as speakers or camera verify the app works well with other applications that may be running at the same time and using the same resources.

Apple is good about announcing upgrades before they are available and often the community has a good read on upcoming updates and upgrades. This can provide help during test phases to ensure that the latest products and OS versions are supported by a newly published application. It is frustrating to developers and to users when a newly published application is closely followed by an OS upgrade that breaks core functionality.