NPS-AM-11-197

ACQUISITION RESEARCH SPONSORED REPORT SERIES

Apple App Store as a Business Model Supporting U.S. Navy Requirements

25 October 2011

by

Lt. Col. (Ret.) Brad R. Naegle, Senior Lecturer, and

Dr. Douglas E. Brinkley, Senior Lecturer

Graduate School of Business & Public Policy

Naval Postgraduate School

Prepared for: Naval Postgraduate School, Monterey, California 93943

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

The research presented in this report was supported by the Acquisition Chair of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

To request Defense Acquisition Research or to become a research sponsor, please contact:

NPS Acquisition Research Program
Attn: James B. Greene, RADM, USN, (Ret.)
Acquisition Chair
Graduate School of Business and Public Policy
Naval Postgraduate School
555 Dyer Road, Room 332
Monterey, CA 93943-5103
Tel: (831) 656-2092
Fax: (831) 656-2253
E-mail: jbgreene@nps.edu

Copies of the Acquisition Sponsored Research Reports may be printed from our website www.acquisitionresearch.net

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Abstract

Naval Open Architecture (NOA) is the confluence of business and technical practices yielding modular, interoperable systems that adhere to open standards with published interfaces. This approach significantly increases opportunities for innovation and competition, enables re-use of components, facilitates rapid technology insertion, and reduces maintenance constraints. A key enabler of the NOA initiative is the Software Hardware Asset Reuse Enterprise (SHARE) repository. The repository was created in August 2006 to facilitate the reuse of software and thereby reduce future development costs. The total benefit of the repository will correspond to the quality and quantity of the applications deposited into it. Indisputably, the most successful software repository in the public sector is the Apple App Store. As of October 2011, Apple lists more than 425,000 applications available. The purpose of this research is to examine the business model of the App Store to identify which of its effective business practices might be applicable to the SHARE repository.

Keywords: Naval Open Architecture (NOA), software reuse, app store, software acquisition

THIS PAGE INTENTIONALLY LEFT BLANK

About the Authors

Brad R. Naegle, Lieutenant Colonel, U.S. Army (Ret.), is a Senior Lecturer and Academic Associate for Program Management Curricula at the Naval Postgraduate School, Monterey, California. While on active duty, LTC (Ret.) Naegle was assigned as the Product Manager for the 2 ½-ton Extended Service Program (ESP) from 1994 to 1996 and served as the Deputy Project Manager for Light Tactical Vehicles from 1996 to 1997. He was the 7th Infantry Division (Light) Division Materiel Officer from 1990 to 1993 and the 34th Support Group Director of Security, Plans, and Operations from 1986 to 1987. Prior to that, LTC (Ret.) Naegle held positions in Test and Evaluations and Logistics fields. He earned a master's degree in systems acquisition management (with distinction) from the Naval Postgraduate School and an undergraduate degree from Weber State University in economics. He is a graduate of the Command and General Staff College, Combined Arms and Services Staff School, and Ordnance Corps Advanced and Basic Courses.

Brad R. Naegle
Graduate School of Business and Public Policy
Naval Postgraduate School
Monterey, CA 93943-5000
Tel: 831-656-3620
Fax: (831) 656-3620
E-mail: bnaegle@nps.edu

Douglas E. Brinkley is a Senior Lecturer and Director of Instructional Technology for the Graduate School of Business and Public Policy at the Naval Postgraduate School, Monterey, California. He is also a retired U.S. Navy Supply Corps Officer with a subspecialty in computer systems management. During his last two tours on active duty, he served as Force Information Systems Officer for Commander, U.S. Naval Air Forces Atlantic Fleet and as Officer In Charge of the DISA Information Processing Center, Guam. Dr. Brinkley earned his bachelor's degree in economics from Excelsior College, a Master of Science in Information

Systems from the Naval Postgraduate School, and a Doctor of Education in Instructional Technology from Nova Southeastern University.

Dr. Douglas E. Brinkley
Senior Lecturer
Director of Instructional Technology
Graduate School of Business and Public Policy
Naval Postgraduate School
Monterey, CA 93943-5000
Tel: 831-656-2771
E-mail: brinkley@nps.edu

NPS-AM-11-197

ACQUISITION RESEARCH SPONSORED REPORT SERIES

Apple App Store as a Business Model Supporting U.S. Navy Requirements

25 October 2011

by

Lt. Col. (Ret.) Brad R. Naegle, Senior Lecturer, and

Dr. Douglas E. Brinkley, Senior Lecturer

Graduate School of Business & Public Policy

Naval Postgraduate School

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

I. Introduction

This research analyzes Apple's industry-leading expertise using the App Store approach for developing applications for its devices as a possible business model that would benefit the U.S. Navy. The goal of this research is to understand how the U.S. Navy might use Apple's App Store business processes to establish similar processes that have the potential to leverage innovative application development from a wide variety of trusted sources, providing maximum benefits to naval entities and personnel while simultaneously ensuring the safety and security of operations and personnel.

A. Background

Apple's App Store concept is being researched as an approach to help facilitate the Naval Open Architecture (NOA). NOA is the confluence of business and technical practices yielding modular, interoperable systems that adhere to open standards with published interfaces. This approach significantly increases opportunities for innovation and competition, enables re-use of components, facilitates rapid technology insertion, and reduces maintenance constraints. NOA delivers increased warfighting capabilities in a shorter time at reduced cost. A key enabler of the NOA initiative is the Software Hardware Asset Reuse Enterprise (SHARE) repository. The repository was created in August 2006 to facilitate the reuse of software and thereby reduce future development costs. The total benefit of the repository will correspond to the quality and quantity of the applications deposited into it. Indisputably, the most successful software repository in the public sector is the Apple App Store. The purpose of this research is to examine the business model of the App Store and identify which, if any, of its effective business practices would be applicable to the SHARE repository.

B. The Apple App Store

The Apple App Store is extremely popular for both users and developers alike. A testament to this is the phenomenal number of applications that have been registered for distribution. As of October 2011, Apple's corporate website states that the App Store is "the world's largest collection of mobile apps—425,000 and counting in practically every category" (Apple Corporation, n.d.e). The number of applications being offered continues to grow each month. From March 2011 to October 2011 the number of applications increased from 350,000 to 425,000. In July 2010, Juniper Research published a report that estimates that the annual number of app downloads is expected to rise from less than 2.6 billion in 2009 to more than 25 billion in 2015 (Juniper Research, 2010). Figure 1 illustrates how the App Store facilitates the interrelationship between the customer and developer networks, which are also often referred to as separate ecosystems (Hinchcliffe, 2010). The success and growth of one ecosystem feeds the mutual proliferation of the other.

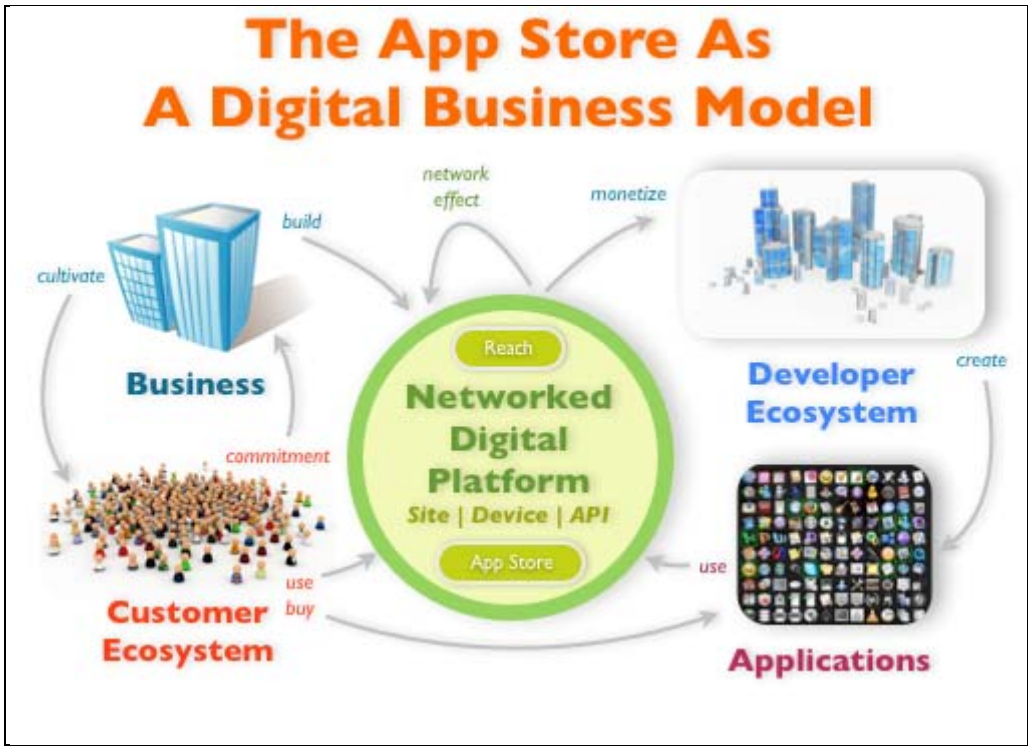


Figure 1. App Store Business Model

Apple's innovative approach vastly increases the number and types of applications that might appeal to their user base. Useful or entertaining applications can be built—under specific business rules, terms, conditions, protocols, and standards—and, after vetting by Apple, be available for sale or free download by Apple device users around the world. Games or applications that may never even be thought of by in-house developers are now being developed and offered to eager Apple customers. Of course, there is a downside to this innovative approach. Safeguards must be in place and enforced to counter malicious applications, viruses, spam, phishing, or even applications that do not work well or that are extremely slow. In addition, this business model invites criticism from those developers denied access, accusations of favoritism or other discriminatory actions, and in the worst case, lawsuits. This research addresses these opportunities and challenges and proposes what the Navy might do to mitigate the downside.

As all of the products become available on the Apple App Store, the company at least appears to endorse the products (despite disclaimers to the opposite) and must take active steps to ensure that there is little risk of customer backlash because of a malicious application or perceived “bad” product. Recognizing how the applications could impact their reputation, Apple has taken great care in controlling the development environments, prescribing the tools and resources to be used, and vetting all potential App Store products before allowing them to be available.

II. Apple's Business Model

During the first phase of this research we contacted Apple's corporate headquarters in Cupertino, California, to gain a better understanding of how the App Store works. We spoke with Apple's Western Region manager, who was very supportive of our research mission, but he was not at liberty to reveal specific details about in-house policies and operation of the App Store. Instead, he suggested that a very thorough understanding could be gained by exploring the operation of the App Store through a developer's perspective. Becoming a software developer for the App Store and documenting that process became the primary methodology to acquire the information needed for this research.

A. App Store Application Development

The first step to becoming an App Store developer is to access the developer introduction website located at http://developer.apple.com. This developer homepage is the gateway to the Apple Development Centers: iOS Dev Center, Mac Dev Center, and Safari Dev Center. The site is well organized and intuitive to use. It is also an effective marketing tool, which encourages developer participation.

Figure 2 is a screen capture of the web page designed to welcome developer participation with headings like "Learn Why You'll Love to Develop with Apple Technologies." Clicking on that heading takes users to another page that expands on the welcome theme with the opening statement:

> Apple provides a complete ecosystem for developers. All the components including hardware, the operating systems, and the developer tools are designed by one company, and they're all designed to work together seamlessly—creating an easier, more intuitive experience so developers can focus on making great apps. (Apple Corporation, n.d.d)

This statement highlights the fact that the hardware, operating systems, and developer tools are all built by Apple. This closed-loop development process is in

stark contrast with the open architecture philosophy of Apple's major software competitor in the portable device market, Android. Because Apple is solely responsible for all aspects of their hardware and software environment, their level of support to users and developers must accordingly be as effective as possible.

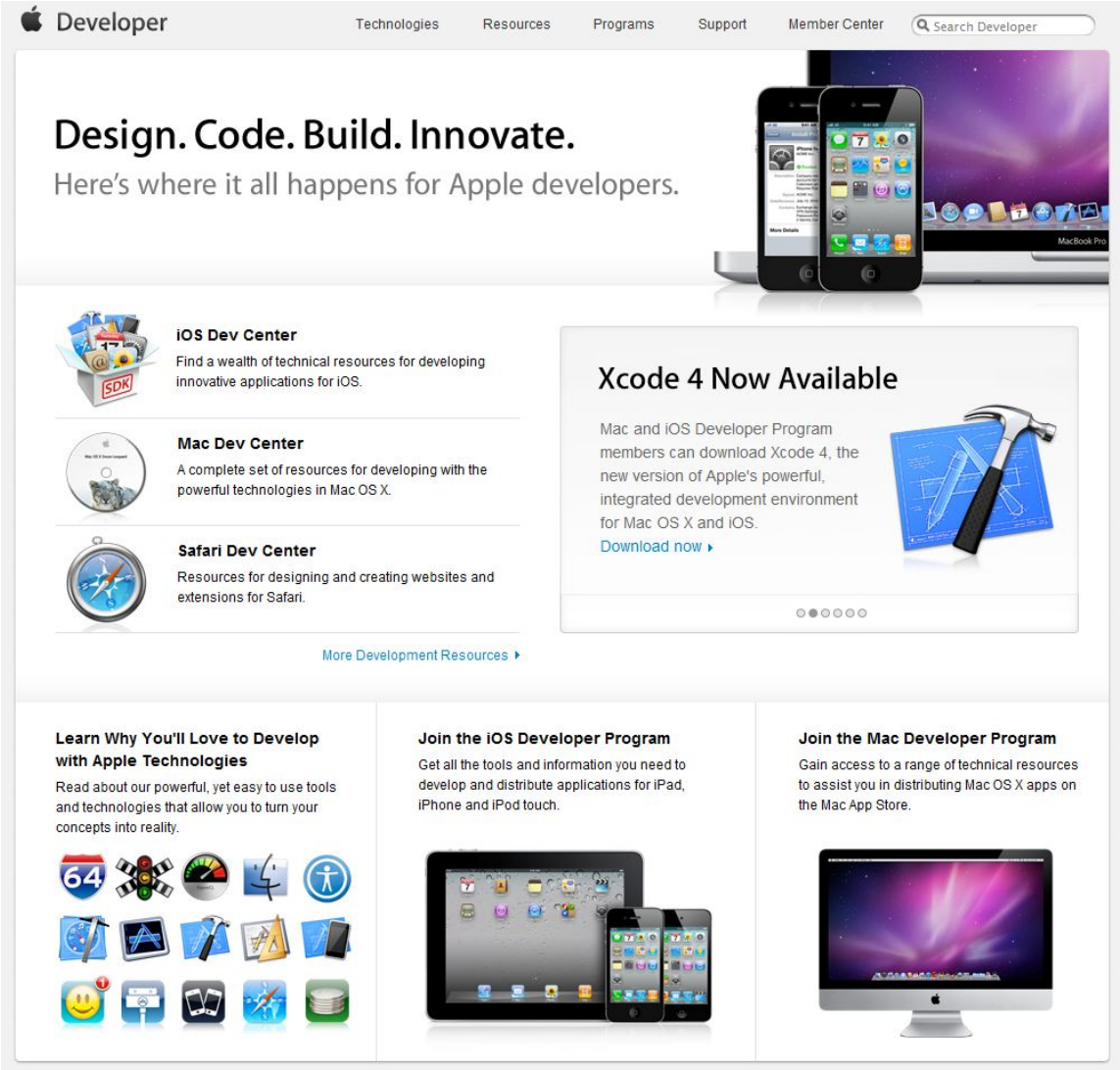


Figure 2. Apple Developer Welcome Page
(Apple Corporation, n.d.d)

It is the iOS Dev Center that is specific to creating applications for the iPad, iPhone, and iPod Touch, which are the devices supported by the App Store. Clicking on the "Join the iOS Developer Program" heading takes the user to

http://developer.apple.com/programs/ios. Figure 3 is a screen capture from this page that depicts the seemingly simple three-step method of developing, testing, and distributing applications on the App Store. Each of the three sections provides a "Learn more" link to give the user additional information.

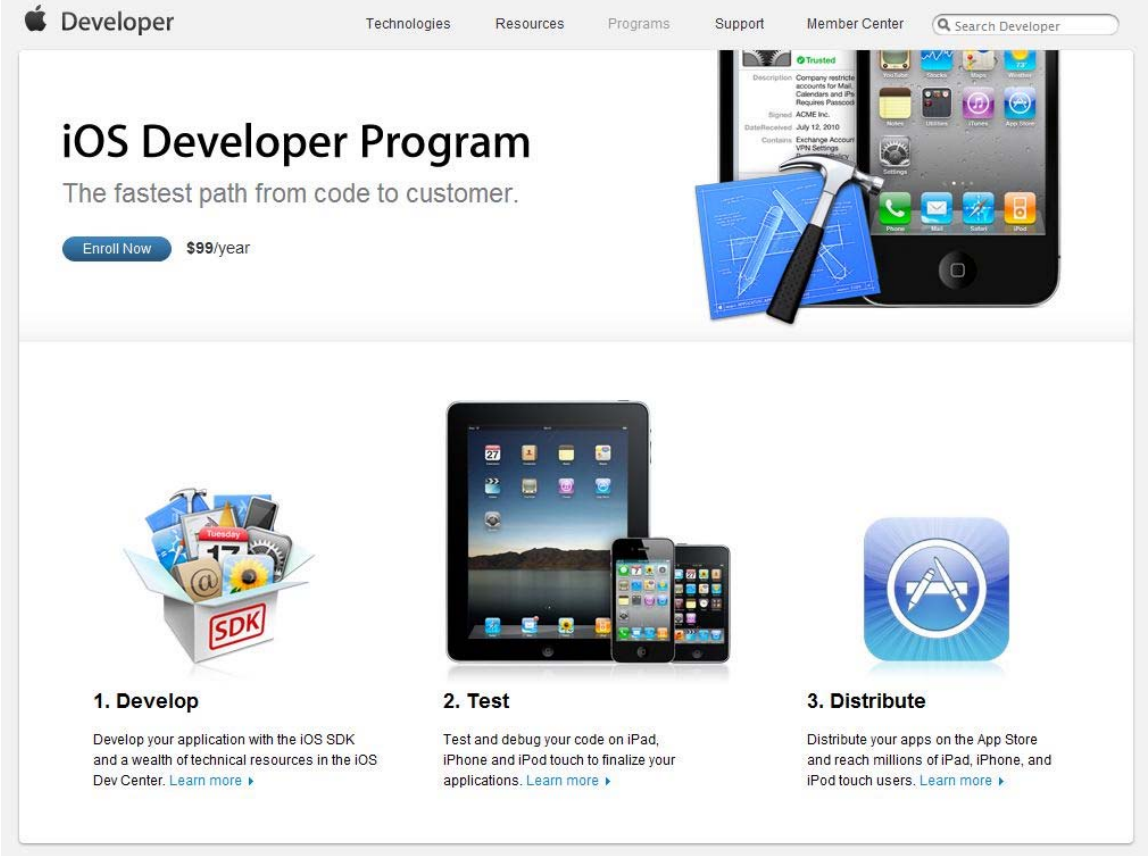


Figure 3. Development Process for iOS
(Apple Corporation, n.d.f)

B. Becoming a Registered App Store Developer

Apple offers the following five different iOS developer programs:

- **iOS Developer Program—Individual:** This is the program used by individual developers wishing to develop apps that will be distributed via the App Store.

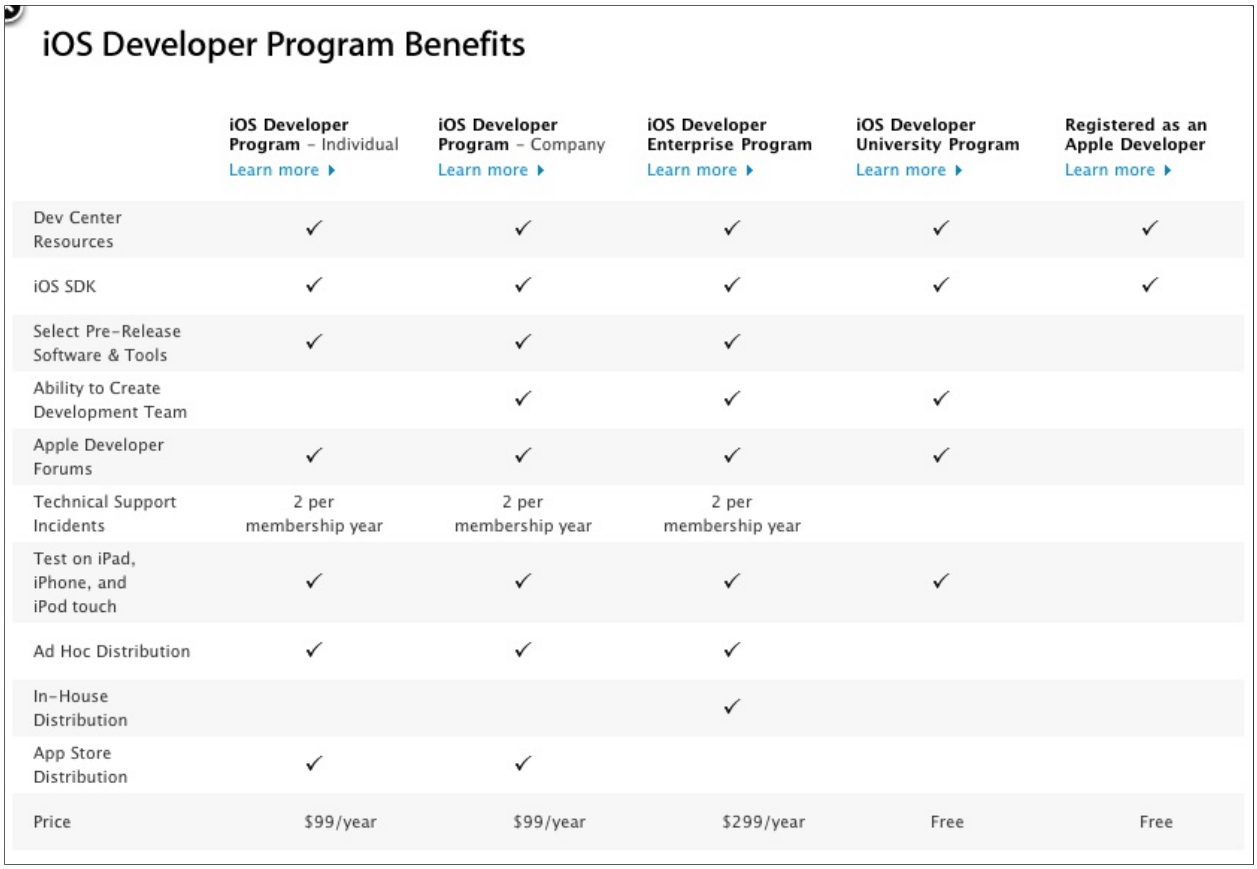
- **iOS Developer Program—Company:** This program is similar to the individual developer program but also allows for the creation of a development team.

- **iOS Developer Enterprise Program:** A special program for organizations wishing to create proprietary in-house iOS apps. Applications created under this program are not distributed through the App Store but rather just to members of the organization.

- **iOS Developer University Program:** A program for higher education institutions to introduce iOS application development into their curriculum. Apps are not distributed via the App Store but can be shared by instructors and up to 200 students within the same development team.

- **Registered as an Apple Developer:** A free program designed to introduce new developers to the iOS coding tools including the SDK. It does not allow for distribution of applications.

Figure 4 lists the different features of each program.



iOS Developer Program Benefits

	iOS Developer Program – Individual Learn more	iOS Developer Program – Company Learn more	iOS Developer Enterprise Program Learn more	iOS Developer University Program Learn more	Registered as an Apple Developer Learn more
Dev Center Resources	✓	✓	✓	✓	✓
iOS SDK	✓	✓	✓	✓	✓
Select Pre-Release Software & Tools	✓	✓	✓		
Ability to Create Development Team		✓	✓	✓	
Apple Developer Forums	✓	✓	✓	✓	
Technical Support Incidents	2 per membership year	2 per membership year	2 per membership year		
Test on iPad, iPhone, and iPod touch	✓	✓	✓	✓	
Ad Hoc Distribution	✓	✓	✓		
In-House Distribution			✓		
App Store Distribution	✓	✓			
Price	$99/year	$99/year	$299/year	Free	Free

Figure 4. iOS Developer Program Benefits
(Apple Corporation, n.d.i)

The authors chose the Individual iOS Developer Program to gain the greatest understanding of the features available to developers wishing to post and distribute applications via the App Store. At the beginning of the on-line registration process, Apple advises the user of the following technical requirement: "You must have an intel-based Mac running Mac OS X Snow Leopard to develop Mac OS X and iOS apps for the App Store" (Apple Corporation, n.d.g).

The requirement that iOS applications can only be developed from within Apple's proprietary OS X operating system emphasizes Apple's tight control over the entire process.

During the first step of registration, users are given the option of creating a new account for the developer program or of using an existing Apple ID such as that

used by an iPod or iPhone user when they registered to access the App Store. The next step asks the user to choose between enrolling as an individual or a company. After selecting individual, the user is presented with a request to update their profile by answering the following questions:

- Which Apple platforms do you develop with?
- What is your primary market?
- Which types of iOS applications do you plan on developing?
- Please select the primary category for your application(s).
    - Free Applications
    - Commercial Applications
    - Enterprise (In-House) Applications
    - Web Applications
- How many years have you been developing on Apple Platforms?
- Do you develop on other mobile platforms? (Apple Corporation, n.d.b)

The next step after creating/updating your profile is to agree to Apple's extensive Apple Developer Agreement. This is a long scrolling document that the author assumes few users ever read. The website does not require scrolling through the document prior to clicking on the "I Agree" button. The next screen is used to enter the developer's credit card identity information for charging the required $99/year fee. Actual payment of the fee is handled by putting the developer program registration request into a shopping cart and then being redirected to Apple's online store to complete the transaction. Upon checking out, the developer is advised that the registration may take up to 24 hours to process. Receipt of the actual developer welcome e-mail was within one hour of the 24-hour estimate.

The developer registration welcome e-mail is very short. It basically says "thank you for joining the iOS Developer Program" and gives you a link to log into the Developer Member Center. Figure 5 is a screenshot of the Member Center.

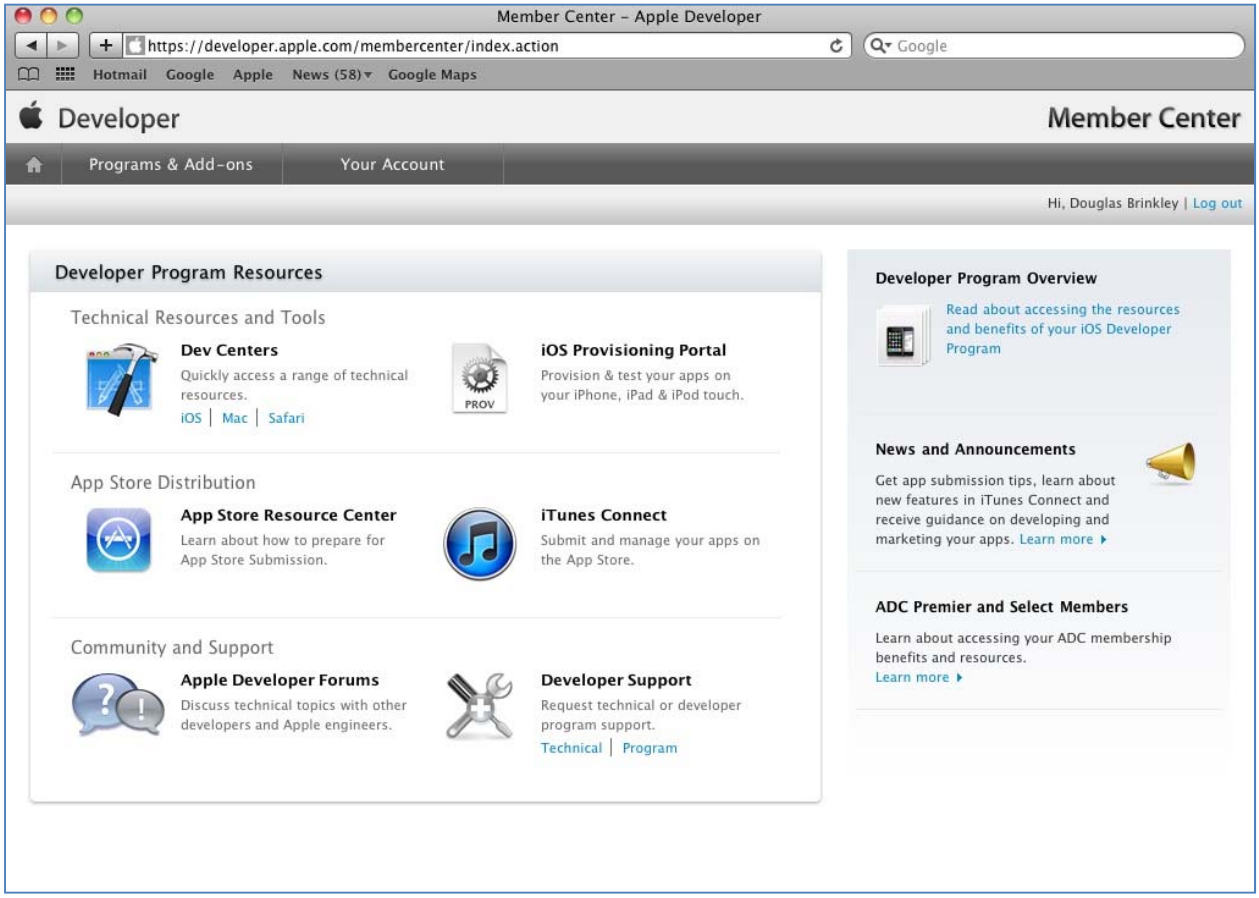


Figure 5. Developer Member Center
(Apple Corporation, n.d.a)

Clicking on the iOS link under the Dev Centers heading takes the user to a much more comprehensive page targeted at getting the developer started on building applications. The main section on the iOS Development page is titled Resources for iOS 4.3, and it includes links to the following items:

- **Downloads**—Link to several items including the iOS SDK.
- **Getting Started Videos**—Watch Apple experts discuss a range of introductory concepts for iOS development.

- **Getting Started Documents**—Learn the fundamental concepts and best-practices for iOS development.
- **iOS Developer Library**—Select from a range of technical documentation on iOS development.
- **iAd JS Reverence Library**—Select from a range of technical documentation on developing with iAd JS.
- **Coding How-To's**—Learn how to incorporate features of iOS in your application.
- **Sample Code**—Use these samples to inspire development of your own great applications.
- **Apple Developer Forums**—Discuss iOS development with other developers and Apple engineers. (Apple Corporation, n.d.h)

These links are only about one third of the resources available on the iOS Dev Center web page. There were several other sections including the App Store Resource Center, News and Announcements, iOS Developer Opportunities, and News and Announcements. As stated earlier, the iOS Dev Center portal is very comprehensive and serves as an effective one-stop-shopping source for just about any topic applicable to App Store development.

C. The iOS Software Development Kit (SDK)

Perhaps the greatest resource made available to new developers is the iOS Software Development Kit (SDK). The major components of the SDK include the following:

- **Xcode**—This refers to Apple's complete integrated development environment (IDE), which integrates all of the SDK's features: the code editor, the build system, the graphical debugger, and project management.
- **iOS Simulator**—Run, test, and debug your application locally on your Mac using a simulated iPhone and iPad.
- **Instruments**—Collect, display, and compare performance data graphically in real-time to optimize your application.

- **Interface Builder**—Interface Builder makes designing a user interface as easy as drag and drop.

In addition to the iOS SDK, Apple describes their other development resources as follows:

- **Apple Developer Forums**—Post iOS SDK development topics and questions for an open discussion with other iOS developers and Apple engineers.
- **Getting Started Videos and Documents**—Covers a range of topics from tools and frameworks to development best-practices and design methods.
- **iOS Reference Library**—A rich collection of documentation, guides, and articles categorized so users can quickly find the information you're looking for.
- **Coding Resources**—Inspire your own development with a library of sample code and coding how-to's. Use these examples to add new functionality or to enhance your application's current design.

D. The Software Development Process

The tools in the SDK support a development process designed to encourage new developers and make them feel comfortable with the iOS environment. They allow you to quickly get a prototype user interface up and running to see what it will look like. You can add code a little at a time and then run it after each new addition to see how it works. The following is an overview of the general process:

1. Use Xcode to select a project template. The templates allow you to get off to a fast start, after which you add your unique code and more features.
2. Design and create the user interface using Interface Builder. Interface Builder has graphic design tools you can use to create your app's user interface. These tools save the developer time and effort from having to design the interface from scratch.
3. Write the app's unique code using the Xcode editor.
4. Run your app on your Mac using the iPad/iPhone Simulator. This is an excellent tool for quickly testing new iterations of your app but there

are many real-world characteristics that the simulators cannot adequately test, such as processing speed.

5. Because of the limitations of the simulators, you must also test your apps on a real iPad/iPhone. Apple provides Development Certificates that allow apps under development to be loaded on specific devices for the purpose of testing.

E. App Store Submission Process

After developers test their applications using the software simulator and again on appropriate testing devices, they are ready to submit the applications to Apple for final testing and review. The only two formats Apple accepts are .zip and .ipa files (e.g., <AppName>.zip or <AppName>.ipa). These must be 2 GB or less in size. Apple calls these files the *binary* package, and they contain everything needed to install the application on a device, as well as the app icons required for the App Store. Applications are submitted via iTunes Connect (http://itunesconnect.apple.com). Table 1 gives a list of the information and files needed during the submission process.

Table 1. Data and Files Needed for Submission to the App Store

Item	Description	Limits
Application Name	The name of the application	None, but it will be shortened as required
Application Description	The content displayed in the App Store listing	4,000 characters maximum
Keywords	The words, features, and functions used to describe your application. Apple will use these words to associate your app with user searches in the App Store.	100 characters max; separated by commas
Application Binary	The App Store version of the application (iPhone, iPad, or Universal)	Must be .zip or .ipa; 2 GB or less
Large Icon Image	The large version of the app icon that will appear in the App Store	At least 512 x 512 pixels; at least 72 dpi
Screenshots	Up to five screenshot images	Must be a .jpeg, .jpg, .tif, .tiff, or .png

For paid apps, developers must also state what price tier they want their app to be sold. Apple uses a price tier system ranging from Tier 1 at $0.99 to Tier 85 at $999.99. The tiers generally increment by $1, but in the upper tiers they increase by $5, $10, $50, and eventually by $100. Developers collect 70% of the selling fees and Apple retains the other 30%.

After the application has been submitted, there are 16 possible status categories:

- **Waiting for Upload**—Appears when developers have completed entering their metadata but have not finished uploading the binary or have chosen to upload the binary at a later time. The app must be in the "waiting for upload" state before the binary can be submitted through the Application Loader.
- **Prepare for Upload**—Appears when the developer has created a new version, but has not yet clicked the Ready to Submit Binary button.

This state also indicates that the developer can now deliver the binary through the Application Loader.

- **Upload Received**—Appears when the binary has been received through the Application Loader, but is still being processed by the iTunes Connect system.

- **Invalid Binary**—Appears when a binary is received through the Application Loader and has been processed, but the binary is invalid. Examples of an invalid binary are as follows: the binary icon does not meet Apple's requirements, the payload directory is at the wrong level in the .app wrapper, the developer attempted to use a non-increasing CFBundleVersion, and so forth.

- **Missing Screenshot**—Appears when the app is missing a required screenshot for iPhone and iPod touch or iPad for your default language app or for your added localizations. At least one screenshot is required for both iPhone and iPod touch and for iPad if you are submitting a universal app.

- **Waiting for Review**—Appears after the developer submits a new application or update and prior to the application being reviewed by Apple. This status means that the app has been added to the app review queue, but has not yet started the review process.

- **Waiting for Export Compliance**—Appears when your Commodity Classification Automated Tracking System (CCATS) is in review with Export Compliance.

- **In Review**—Appears when Apple is reviewing an app prior to the application being approved or rejected. When the status of an application is in review, the developer has the option to reject the binary they have submitted by clicking Reject Binary. This will remove the binary from the review queue and will allow for another update to be submitted. If the binary is rejected, the status of the app will change to "developer rejected" and when the binary is re-submitted, the review process will start over from the beginning.

- **Pending Contract**—Appears when the application has been reviewed and is "ready for sale" but the developer's contracts are not yet in effect. Developers may check the progress of their contracts in iTunes Connect by clicking on the Contracts, Tax & Banking information module.

- **Pending Developer Release**—Appears when the version of the app has been approved by Apple and the developer has turned on the

Version Release Control, but has not yet clicked Send Version Live. There will also be a pending action symbol on the version. The version will remain in this state, and thus will not be live on the App Store until the developer clicks Send Version Live.

- **Processing for App Store**—Appears when the version is being processed to go live on the App Store. Once the processing is complete, the version state will change to "ready for sale." This is a temporary state (approximately 1–2 hours).
- **Ready for Sale** —Appears once the application has been approved and posted to the App Store. When the application is in this status, the submitter has the option to remove it from the store by going to the Rights and Pricing page and removing all App Store territories.
- **Rejected**—Appears when the binary has been rejected.
- **Removed from Sale**—Appears when the binary has been removed from the App Store.
- **Developer Rejected**—Appears when the developer has rejected the binary from the review process. Existing versions of the application on the App Store will not be affected by self-rejected binaries in review.
- **Developer Removed from Sale**—Appears when the developer has removed the application from the App Store.

If developers have opted to receive notifications, they will get an e-mail whenever the app changes status. They can also see the current status and status history by clicking on the Status History link within the application's version information screen. The status history will include an audit trail of when the app was submitted, when it went into review, and other changes up to the point it was ready for sale.

F. Apple's Vetting Process

Apple provides a detailed listing of their review guidelines at http://developer.apple.com/appstore/resources/approval/guidelines.html, and many of the guideline items were introduced earlier in this research. The guidelines cite more than 100 different reasons that applications might be rejected. In addition to the reasons associated with technical shortcomings, applications are just as likely to

be rejected for non-adherence to Apple's policies. Noted authors Bove and Goldstein (2011) cited the following as some of the most common policy reasons for application rejection.

- **Linking to private frameworks**—Apple rejects apps that call external frameworks or libraries that contain non-Apple code.
- **Straying too far from Apple's guidelines**—The authors cited examples of their own apps that were rejected simply because their use of menu highlighting was not done in the manner described in Apple's guidelines.
- **Copying Apple's existing functionality**—Although you should use the functionality provided for developers, you should not copy something that Apple already does at the App level. For example, Apple has already developed a Web browser called Safari. Apple would reject other applications that duplicate (compete) with their own app.
- **Apps not providing adequate user feedback**—An example would be an application that requires an internet connection. Apple may reject the app if it does not automatically advise the user when the connection is lost.

During Apple's Worldwide Developers Conference (WWDC) 2010, Steve Jobs stated in his keynote address that apps are rejected for three main reasons: the application does not function as advertised, it uses private APIs (generally, this means creating programming functions in your app that are not available in the iOS SDK), and the app crashes frequently (Yarmosh, 2010). Most new apps are reviewed within seven days. As of May 5, 2011, the Apple developer's website stated that 91% of new apps and 95% of app updates were reviewed within the last seven-day window.

For apps that are rejected for technical reasons, Apple encourages the developer to make the required modifications and resubmit the program for review. The $99 developer sign-up fee includes two Technical Support Incidents (TSI), which Apple defines as follows:

A Technical Support Incident is a benefit of the Mac and iOS Developer Program and allows program members to request code level support from our developer technical support engineers. Your issue will be assigned to an engineer who can help you troubleshoot your code, offer direction to additional technical resources, or provide workarounds that will fast-track your development. (Apple Corporation,n.d.c)

THIS PAGE INTENTIONALLY LEFT BLANK

III. Pros and Cons of the App Store Business Model

As stated at the outset, there can be no arguing that the App Store has been a phenomenal success for Apple. Their management of the App Store is in keeping with the company's proprietary nature for tightly controlled operations. The authors' personal experience with the App Store developer program has led to the following observations regarding the advantages of this business model.

- **Exemplary and comprehensive web portal**—Apple's developer web portal is second to none. It is truly a one-stop shopping experience with the ability to lead even a novice developer through the steps necessary to achieve success. The website's extensive use of multimedia, using a broad range of products from hyperlinked text documents to instructional videos, helps to ensure a clear understanding of important topics.

- **Greatest possible assurance of safe applications**—Apple's rigorous review process and tight control of programming minimizes the adverse risk of viruses, worms, malware, and adware. This inherent safety instills confidence and effectively removes any hesitation end users might have about downloading new applications.

- **A symbiotic relationship between the end user and developer ecosystems**—The success and proliferation of the end user and developer networks feed each other. As more and more users join the Apple customer base, the greater the potential market and attraction for developers to contribute new applications to the App Store.

- **Effective management and marketing of application updates**—Most software goes through a life cycle of continuous improvement as long as the developer is incentivized to continue working on it. The App Store business model encourages continued application development by giving developers mechanisms to advertise new versions of "old favorites" that customers might be interested in.

- **A self-contained application directory and store**—End users only need to access one site to both view the directory of available applications and download the applications. Because both the directory and the store are one, there is no question that what is listed is available.

- **Multi-device compatibility simplifies user access**—The App Store can be accessed by all of Apple's portable devices. This universal access reduces the effort to discover and download new applications to as simple a process as possible and eliminates the need for additional resources (like another computer). This direct download and install capability gives the users instant gratification and thus encourages them to frequent the App Store and to try out new applications.

- **Centralized payment processing**—Apple simplifies the payment process by setting up a single account for each user that is used to pay for any application in the App Store. Apple keeps 30% of each sale as its commission and gives the other 70% to the respective developer. Previous business models for software distributed as "shareware" required the end user to download the program from one source and then pay the developer directly. Purchasing from several different developers meant setting up a separate payment process for each of them.

- **End-user feedback and rating system**—The App Store includes a feature to allow customers to rate and comment on their satisfaction with the applications they have purchased. This public feedback then gives the developer a clear, customer-driven set of criteria to improve against and gives customers the information they need to determine if an application really meets their needs. This is an extremely valuable feature for customers when they are presented with a large number of similar applications. The public feedback system helps to narrow the field to those applications judged best by previous customers.

Whereas the list above outlines several advantages of the Apple App Store business model, the following lists a few of the disadvantages to consider as well.

- **Potential bottleneck for distribution**—Apple has taken some criticism in the past for taking excessive time to approve new applications. Developers are extremely sensitive to delays that could have an adverse impact on their competitive advantage. They understand that one key to their success is being the first to market an app with new functionality. The competitive nature of the business results in a limited lifespan for most applications.

- **Excessive delays in correcting application bugs**—Even if a developer discovers and fixes a bug in an application within 10 minutes of its being posted, Apple still requires a week-long process to review the fix. During this time, users can leave thousands of angry

comments in the App Store which could doom the revised software from even being looked at by future potential customers.

- **You can only download the current version of an application**—Apple routinely updates the operating system on its portable devices. Some applications have compatibility problems with specific versions of the operating system. Once a developer updates an application to run on the most recent version of the operating system, the application may not work for a large number of users who have not updated their device. Apple's policy of not allowing simultaneous versions of applications for different versions of the OS forces developers to abandon some of their customer base.

- **Inability for developers to contact users who leave negative feedback**—The App Store's public feedback system does not allow a developer to communicate with a customer who may have left a negative comment based on erroneous information. For example, as stated previously, some features of an application might only work with the current version of the operating system. If a user has not updated their device then they may leave a review stating that the application is faulty.

- **Rejection of apps that employ non-Apple code**—Perhaps the greatest disadvantage of Apple's proprietary developer policies is the rejection of apps that use code not included in the iOS SDK. Apple is very strict about this and it effectively bars developers from creating apps that use features or functions of Apple's portable devices in a manner not sanctioned by Apple. For example, some early apps were rejected because they gave the user the ability to tether their phone to other computing devices and thus share the iPhone's data connection. Eventually this very popular feature was added to Apple's built-in iOS capability after AT&T ensured that they could detect the use of tethering and charge more for it. Tethering applications are still rejected from the App Store, but now the reason given is that it duplicates a function that is already available. It can be assumed there are many more apps that would also be very popular but are rejected because they provide some functionality not intended by Apple. This type of restriction is one of the primary reasons given by users and developers for switching over to the more open Android platforms.

A. Navy and DoD Business Practices Laws and Regulations

While Apple was free to set its own rules for managing their App Store, there are voluminous laws and regulations designed to control and guide the Department of Defense's (DoD) procurement of materiel, goods, and services that the Navy must contend with. Virtually all of these regulations apply directly to the Navy's business practices, along with their own regulations that supplement laws and DoD regulations, policy, and guidance. One critical area is Information Assurance (IA). Systems that interoperate or interface with specified operational systems will need to be certified under the DoD Information Assurance Certification and Accreditation Process (DIACAP; Under Secretary of Defense for Acquisition, Technology, and Logistics [USD(AT&L)], 2007). The Navy App Store business model must be in compliance with, and will be shaped by, these directing and guiding documents. Whereas the totality of the laws and other guiding documents appears to severely limit the Navy's ability to implement an innovative approach such as Apple's App Store, there is still significant flexibility within the Federal acquisition system for such innovation.

B. Establishing the Requirement

Within the DoD, requirements for goods and services must be established and prioritized. Usually, a DoD user group will establish the "need" for the goods or services, which will be approved and identified with a program, flow up through the Service and the DoD, be prioritized against all other programs, and be forwarded for consideration for funding. After consideration, Congress will provide authorization to expend the funds and appropriate the actual funding at some specified level. This process establishes one of the key DoD financial management tenets: *Bona Fide Need*. The funding provided by Congress is generally applied according to the priority until all funding is allocated, typically leaving many authorized programs and budget items unfunded.

C. Funding

Financial laws, regulations, and guidance are closely linked to the organizations controlling the procurement of materiel, goods, and services. Financial management within the Federal Government is much more complex and restrictive than that of commercial enterprises. The U.S. Congress provides budget authority to federal agencies (including the DoD and the Navy) through a two-step process. First, Congress provides the authority (authorizations law) to expend funding in a specified manner. Second is the appropriation of public funds (appropriations law), providing a specified amount of funding, specifying the time period, and designating the purpose for the obligation of authorized and appropriated funds.

D. DoD and Navy Contracting

Given all of the laws, regulations, and guidance governing procurement and financial management, a contract is typically the vehicle that binds the U.S. Government to expend funds in exchange for goods and services provided by another entity (FAR, 2005, Part 4). The DoD typically contracts with business entities. Individuals desiring to contract with the DoD must establish a business identity and register on the Central Contractor Registrations (CCR) system (FAR, 2005, Part 9). The CCR can be accessed through the U.S. Government website shown in figure 6 (General Services Agency [GSA], 2011). The type of contract awarded is typically selected based on which party must bear the risk of performance. For delivery of mature, low-risk products and services, a fixed-price contract is usually appropriate and the contractor bears the risk of delivery at contracted prices. As the risk of performance increases, the Federal Government must accept more risk to ensure that contractors will attempt to provide the goods or services required. Cost-plus contracts are typically used under these circumstances so that a contractor's risk is minimized, and they remain incentivized to develop the product that the Federal Government is requiring.

Figure 6. Central Contractor Registration Login Page
(GSA, 2011)

E. DoD App Store Experience

The Army and the Air Force have initiated limited steps towards using an App Store approach in their respective services. Concerns over vetting and security continue to be significant challenges in fully using the applications in a military environment. In the June 2011 *National Defense Magazine,* in the article titled "Smartphones-for-Soldiers Campaign Hits Wall as Army Experiences Growing Pains," Sandra Erwin (2011) writes:

> A growing frustration for smartphone proponents is that the very features that make the gadgets so attractive would be neutralized by restrictive security policies. "I need these devices and applications to be CAC enabled," [Deputy Army CIO Maj. Gen. Steven] Smith says, referring to the "common access card" that military and government employees use to log into computers or to enter secure facilities.

> The Army's former CIO [Chief Information Officer], Lt. Gen. Jeffrey Sorensen, garnered the media spotlight when he launched "Apps for the Army" more than a year ago. But the vision is still catching up with the service's onerous regulations. "We need a way that we can vet the applications," Smith says. "If you go to some companies, they can vet applications in hours or days, and it takes us 81 months," he says. "There has to be a happy medium. We need an apps store with TTP's [tactics, techniques, and procedures] that you've already worked." … "We have serious questions about operating and maintaining those apps," Smith says.
>
> The Army plans to launch a new apps store in 2012. The current CIO, Lt Gen. Susan Lawrence, says there will be another "Apps for the Army" challenge that will be open to any developer. … CIO officials now are "designing monetization business models and addressing intellectual property rights," in preparation for the 2012 competition.
>
> The concerns are not unique to the Army. The Air Force also is exploring ways to expand the use of smartphones, says Lt. Gen. William T. Lord, Air Force CIO. … But like the Army, the Air Force worries about troops having possibly too much access to information at their fingertips. "How do we secure data?" Lord asks. "Do we tell them to park their Facebook accounts, their iPhones at the door? What are you going to do 18 months from now when that thing [the phone] is absolutely ancient?" he says. (Erwin, 2011)

While the article specifically references "smartphones," the challenges surrounding the military use of applications developed for widely distributed and interconnected devices remain the same, with the possible exception of the device ("phone," in the article cited previously) becoming "ancient" in 18 months. The question remains: how do you leverage the innovation of the App Store business model, while at the same time ensuring security and complying with all applicable laws, regulations, and directives?

THIS PAGE INTENTIONALLY LEFT BLANK

IV. Apple App Store Concept Analysis

It is clear from the research that Apple has been successful in leveraging innovative products and services using the App Store developmental approach. Apple continues to offer potential application developers a method to develop, refine, and market applications within their defined processes, providing their customers with an extremely wide variety of useful and entertaining products. In addition, the number and variety of the developed applications could not have been produced in-house by the available Apple development staff, and outside developers would not have a method for effectively developing Apple product-ready applications or marketing them without the App Store concept. The App Store business approach is a classic "win-win" scenario.

The goals of Apple and of the App Store developers appear to be significantly met through the App Store concept. Apple is able to provide its customers with a dizzying array of innovative products and the outside developers have a method to develop and market their intellectual properties to Apple's extensive customer base. Those developers who market their applications for profit are especially well-served through the App Store arrangement, and Apple recovers App Store operating costs through their profit-sharing agreement with developers.

The data presented indicate that Apple expends a significant amount of resources to support the App Store concept. The amount of application developer support, detailed previously, is substantial. The production of the developer application and support websites, the software engineering support provided to developers, the extensive product vetting process, the online marketing, and the App Store financial management and reporting resources represent a massive commitment of personnel, time, and funding resources. As Apple continues to offer and promote the App Store process, it is clear that the concept has a positive cost benefit for the company.

Apple must comply with applicable laws and regulations, and beyond those, must protect its corporate image and products. Apple's App Store vetting process is designed to identify and eliminate any product with potential illegal, immoral, or controversial application and, further, any product that changes or adversely affects any Apple software or product. To that end, Apple carefully controls the development process from initiation through implementation, and then continues to monitor applications for user acceptance and any latent problems that may appear after deployment. Developers are restricted to using only Apple software and development tools, which is essential for Apple's testing and vetting processes. This proprietary control of the application development process is in stark contrast to the open approach used by Apple's major competitor, Android.

V. Navy App Store Implementation Analysis

Our research indicates that it may be plausible for the Navy to establish its own App Store concept, if it is able to define and communicate its requirements, control the process, resource its support for the process, and establish effective business rules within the DoD and Navy laws, directives, and guidelines. The products and applications the Navy desires will obviously be quite different from those of Apple, requiring an end-to-end process definition to incentivize innovation, but narrowing the focus to meet well-defined Navy needs.

A. Defining and Communicating Product and Business Requirements

Unlike Apple, which is interested in almost any application that appeals to its customers, the Navy will likely need to define more specifically the types of applications it desires. This need should not overly restrict innovative solutions, but should help potential developers focus on desired areas. For example, the Navy might indicate that they desire applications that assist in "logistics tracking." There may be hundreds of innovative applications that would apply to that broad area, most of which would likely have utility for the Navy. There would be a significant amount of requirements analysis to develop the broad areas before any effective solicitation could be publicized.

Whereas Apple is free to enter into the App Store agreement with both businesses and individuals, the DoD and the Navy typically contract with businesses and corporations for products, imposing laws, policies, regulations, and guidance from all levels of government into the contracting process. Individuals who are not part of any corporation would likely have to establish a business identity and register on the U.S. Government's Central Contractor Registration (CCR) website at URL https://www.uscontractorregistration.com/.

With the Apple App Store concept, profit-seeking developers clearly understand the method for making money and the associated risks for developing an application that is not popular or is perceived to be too expensive. Potential developers for the Navy App Store will need to be similarly knowledgeable on both the potential for profit and the associated risks. Because profit is based on application sales and there is a fee to participate, Apple does not need to limit the number of participating application developers because the corporation is not the source of funding. There will likely be a limit on the available government funds, thus Navy App Store participation would necessarily be limited, too. The government solicitation will need to be very clear regarding requirements, bidding, source selection, and how potential developers will earn the funds. The contract needs to be carefully worded to garner the maximum developer innovation, yet guide the efforts towards the maximum utility for the Navy and provide appropriate levels of control.

B. Controlling the Process

It is evident that Apple tightly controls the application development process from beginning to end, and even characterizes itself as "control freaks." This controlled process is effective for Apple, and appears to provide the same type of controls likely to be needed by the Navy.

Requiring developers to use Apple's software, software tools, and test environments ensures a thorough assessment of the application software by the experts in their own software. It would be difficult for a developer to hide malicious or other undesirable software features from the Apple software engineering staff, and any problems with system or device interoperability can be addressed well before deployment.

The Navy will need varying levels of control over the application development process depending on the classification or sensitivity of the systems being accessed by the applications. In classified or information assurance (IA) environments, the

Navy would likely require significant control and access to source code to properly assess risks and for the DIACAP certification process. The Apple App Store model appears to provide that level of control, and the Navy would necessarily need to provide a similar level of support with software engineers, software tools, testing environments or software integration laboratories, and other development support. If the applications developed were not supporting or interfacing with sensitive or classified systems, a lower level of control over the process and products would appear to be acceptable. However, there would continue to be a need for some level of government-provided support for analyzing proposals and products, as well as for conducting risk analyses.

The government solicitation and contracting processes will need to clearly identify the level of control that will be imposed on potential application developers. Potential Apple App Store developers enter into binding agreements with all of the Apple-imposed controls over the development, content, and interoperability of any proposed application development, and the Navy would need to do something similar.

Identifying contract deliverables, data requirements, data rights, progress payments (if any), government acceptance criteria, and government oversight and control mechanisms will likely be challenging. The U.S. Government acquisition process is controlled through a substantial amount of laws, regulations, and guidance documents, and these will likely add to the challenge of crafting a contract that is compliant yet allows for the innovation sought.

There may be an opportunity to leverage App Store products that already exist, although there would be the obvious limitation of operating on Apple devices exclusively if they are used as developed. If the desired capabilities were known, searching the 350,000 applications could yield positive results. This process may also identify developers with particular skills in developing desired capabilities, and solicitations for Navy-specific development could include those high-potential developers.

C. Vetting the Products

It is clear that the product vetting process is significantly important for Apple's App Store applications and would be critical for the Navy if applications were used operationally or interoperated with networks and other systems. Apple dedicates significant resources for the vetting process and has controlled the development process to help facilitate the vetting. From Erwin's *National Defense* article cited earlier, it is evident that the Army has had significant challenges in properly vetting applications, taking up to 81 days to do so (Erwin, 2011).

Navy App Store vetting processes would necessarily be required to be established and resourced prior to any solicitation because developers would need to know the process for qualification and acceptance of developed applications. The vetting process would likely include security, IA, vulnerability analysis, and other critical issues for military operations.

VI. Conclusions and Recommendations

The Apple App Store process has been an effective way for Apple to attract innovative application development and provide its customers with an extensive array of products beyond the company's ability to create internally. Implementing the Apple business model, the Navy would need to focus potential developers on areas that it considers most beneficial.

Setting the Navy requirements for application development is an essential step in initiating the App Store business model. Establishing the requirements would also define how and at what level the applications would be operated and the systems that need interoperability or interfaces. Defining the requirements would help determine the need for IA and other classified or sensitive systems considerations involved with the application development and help determine the necessary associated level of control over the development process and delivered products.

Apple's tight control of the development software and procedures is a key enabler of their vetting process. An open-source model would significantly complicate the control and vetting processes. The solicitation and contract will need to specify development controls that will be implemented.

Vetting or qualifying the products will likely entail significant effort and resources. Depending on the intended use of the applications, the vetting may require varying degrees of analysis and testing, DIACAP certification, or other risk-based analysis and testing for use on classified and non-classified systems. The degree of control over the development process would have an effect on the difficulty in the vetting process. The more that is known about the software, engineering tools used, and testing regiments, the easier it is to complete the vetting process, which is the control philosophy used by Apple for App Store development.

Significant resources are required to research and establish the requirements, provide the solicitation, process and select potential developers, control the process, vet the products, and support the fielding of successful applications. Apple would not divulge the actual number of people or the cost associated with the App Store model, but it was clear that the investment in personnel, company resources, and time was substantial. Resourcing this level of effort would require a significant investment by the Navy.

The Navy may be able to leverage Apple App Store products that have already been developed and are currently available. Although they have been built to support Apple's i-devices, it may be cost effective to purchase the appropriate equipment and buy the applications directly. If the Navy does not want to limit the hardware to Apple products, desirable applications might be contracted and redeveloped by the same expert or experts in a form that meets the Navy's criteria.

References

Apple Corporation. (n.d.a). Apple developer member center. Retrieved from http://developer.apple.com/membercenter/index.action

Apple Corporation. (n.d.b). Apple developer program enrollment. Retrieved from https://connect.apple.com/cgi-bin/WebObjects/register.woa/50/wo/oFwdA51mLj4QqgAdFMz94g/0.0.1.3.5.7

Apple Corporation. (n.d.c). Apple developer support center. Retrieved from http://developer.apple.com/support/technical/

Apple Corporation. (n.d.d). Apple developer welcome page. Retrieved from http://developer.apple.com/

Apple Corporation. (n.d.e). Apps for iPhone. Retrieved from http://www.apple.com/iphone/apps-for-iphone/

Apple Corporation. (n.d.f). Development process for iOS. Retrieved from http://developer.apple.com/programs/ios

Apple Corporation. (n.d.g). Enroll in the Apple developer program. Retrieved from http://developer.apple.com/programs/start/standard/

Apple Corporation. (n.d.h). iOS developer center. Retrieved from http://developer.apple.com/devcenter/ios/index.actionApple Corporation. (n.d.i). iOS developer program benefits. Retrieved from http://developer.apple.com/programs/ios/includes/compare.html#compare

Apple Corporation. (n.d.j). Why you'll love to develop with Apple technologies. Retrieved from http://developer.apple.com/technologies/

Bove, T., & Goldstein, N. (2011). *iPad application development for dummies* (2nd ed.). Hoboken, NJ: Wiley.

Erwin, S. I. (2011, June). Smartphones-for-soldiers campaign hits wall as Army experiences growing pains*. National Defense Magazine.* Retrieved from http://www.nationaldefensemagazine.org/archive/2011/June/Pages/Smartphones-for-SoldiersCampaignHitsWallasArmyExperiencesGrowingPains.aspx

Federal Acquisition Regulation (FAR), 48 C.F.R. ch. 1 (2005).

General Services Agency (GSA). (2011). Central contractor registration (CCR) welcome page. Retrieved from https://www.bpn.gov/ccr/default.aspx

Hinchcliffe, D. (2010, January 21). Enterprise Web 2.0 [Web log post]. Retrieved from http://www.zdnet.com/blog/hinchcliffe/the-app-store-the-new-must-have-digital-business-model/1172

Juniper Research. (2010, July). Mobile App Store downloads to reach 25 billion by 2015 [Press release]. Retrieved from http://juniperresearch.com/viewpressrelease.php?pr=195

Under Secretary of Defense for Acquisition, Technology, and Logistics (USD[AT&L]). (2007, November). *DoD information assurance certification and accreditation process* (*DIACAP*; DoD Instruction 8510.01). Retrieved from http://www.dtic.mil/whs/directives/corres/pdf/851001p.pdf

Yarmosh, K. (2010). *App savvy.* Sebastopol, CA: O'Reilly Media.

2003 - 2011 Sponsored Research Topics

Acquisition Management

- Acquiring Combat Capability via Public-Private Partnerships (PPPs)
- BCA: Contractor vs. Organic Growth
- Defense Industry Consolidation
- EU-US Defense Industrial Relationships
- Knowledge Value Added (KVA) + Real Options (RO) Applied to Shipyard Planning Processes
- Managing the Services Supply Chain
- MOSA Contracting Implications
- Portfolio Optimization via KVA + RO
- Private Military Sector
- Software Requirements for OA
- Spiral Development
- Strategy for Defense Acquisition Research
- The Software, Hardware Asset Reuse Enterprise (SHARE) repository

Contract Management

- Commodity Sourcing Strategies
- Contracting Government Procurement Functions
- Contractors in 21st-century Combat Zone
- Joint Contingency Contracting
- Model for Optimizing Contingency Contracting, Planning and Execution
- Navy Contract Writing Guide
- Past Performance in Source Selection
- Strategic Contingency Contracting
- Transforming DoD Contract Closeout
- USAF Energy Savings Performance Contracts
- USAF IT Commodity Council
- USMC Contingency Contracting

Financial Management

- Acquisitions via Leasing: MPS case
- Budget Scoring
- Budgeting for Capabilities-based Planning
- Capital Budgeting for the DoD
- Energy Saving Contracts/DoD Mobile Assets
- Financing DoD Budget via PPPs
- Lessons from Private Sector Capital Budgeting for DoD Acquisition Budgeting Reform
- PPPs and Government Financing
- ROI of Information Warfare Systems
- Special Termination Liability in MDAPs
- Strategic Sourcing
- Transaction Cost Economics (TCE) to Improve Cost Estimates

Human Resources

- Indefinite Reenlistment
- Individual Augmentation
- Learning Management Systems
- Moral Conduct Waivers and First-term Attrition
- Retention
- The Navy's Selective Reenlistment Bonus (SRB) Management System
- Tuition Assistance

Logistics Management

- Analysis of LAV Depot Maintenance
- Army LOG MOD
- ASDS Product Support Analysis
- Cold-chain Logistics
- Contractors Supporting Military Operations
- Diffusion/Variability on Vendor Performance Evaluation
- Evolutionary Acquisition
- Lean Six Sigma to Reduce Costs and Improve Readiness

- Naval Aviation Maintenance and Process Improvement (2)
- Optimizing CIWS Lifecycle Support (LCS)
- Outsourcing the Pearl Harbor MK-48 Intermediate Maintenance Activity
- Pallet Management System
- PBL (4)
- Privatization-NOSL/NAWCI
- RFID (6)
- Risk Analysis for Performance-based Logistics
- R-TOC AEGIS Microwave Power Tubes
- Sense-and-Respond Logistics Network
- Strategic Sourcing

Program Management

- Building Collaborative Capacity
- Business Process Reengineering (BPR) for LCS Mission Module Acquisition
- Collaborative IT Tools Leveraging Competence
- Contractor vs. Organic Support
- Knowledge, Responsibilities and Decision Rights in MDAPs
- KVA Applied to AEGIS and SSDS
- Managing the Service Supply Chain
- Measuring Uncertainty in Earned Value
- Organizational Modeling and Simulation
- Public-Private Partnership
- Terminating Your Own Program
- Utilizing Collaborative and Three-dimensional Imaging Technology

A complete listing and electronic copies of published research are available on our website: www.acquisitionresearch.net

THIS PAGE INTENTIONALLY LEFT BLANK

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CALIFORNIA 93943

www.acquisitionresearch.net