

Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

National Security Cyberspace Institute, Inc. (NSCI)

Through the combination of research and education, NSCI supports public and private clients aiming to increase cyberspace awareness, interest, knowledge, and/or capabilities. NSCI is committed to helping increase security in cyberspace whenever and wherever possible. NSCI publishes a bi-weekly newsletter ([CyberPro](#)), has published numerous [whitepapers](#) on various cyberspace topics, maintains an [online cyber reference library](#), and has established an [email distribution list](#) for sharing cyber-related resumes to interested parties. NSCI is a small, veteran-owned business headquartered in Virginia.

Questions to Answer

- What are key components that make the Apple iPhone application development model work?
- What would be required / recommended for someone else to emulate this model (e.g. DoD adopt/adapt the model for JC2 Application Development)?

Overview

Apple provides core / basic applications bundled with the iPhone. These include a web browser (Safari), basic email client, maps, phone, video, music and others. Additional functionality is available via applications that can be downloaded / purchased from the [iPhone Applications "store"](#). These additional applications are developed by 3rd party software developers and range in price from 0.99 to more than \$50.

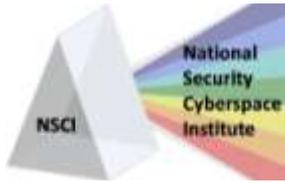
Third party developers who are members of the [Apple Developer Connection](#) may create and submit products for possible listing on Apple iPhone developer and marketing web pages. To be considered, a submission must meet the following guidelines¹.

- You must agree to the iPhone Web Application Submission Agreement for all materials submitted for consideration.
- Your submission must conform to the [iPhone web development guidelines](#) in effect at the time of your submission.
- Your submission should have a complete feature set (stable releases/demos preferred to beta sw)
- Your submission (and the content displayed in or through your submission) must not violate or infringe the intellectual property rights (including trademark rights) of others.
- You must either own all rights to your submission and the content displayed in or through the submission or have written authorization from the owner(s) thereof.
- You must provide at least one working link to access your submission.

By submitting applications for consideration to the iPhone Web Application Submission page, you understand your submission is subject to review by Apple. Apple reserves the right to omit, edit, or reject submissions.

March / April 2008, the App Store's original 800 offerings resulted in over 10 million downloads in its first weekend of operation and achieved \$500m (£310m) in sales during its first month. By October, the Store had grown to 5,550 apps and 200 million downloads. In December its 10,000 apps grew that total to 300 million downloads. January 2009 saw 15,000 apps and 500 million downloads

¹ iPhone Web Application Submission; accessed 12 Dec 2009; <https://daw.apple.com/cgi-bin/WebObjects/DSAuthWeb.woa/wa/login?appIdKey=D534F5C413E680ACBC861EE8025883117C9705F90893AF782751628F0291F620&path=/iphone/index.php>



Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

April, the Store held 35,000 apps and one billion downloads.
July, the App Store's collection easily exceeds 56,000 apps. Add to that the over "inactive" 4,000 apps and we're talking over 60,000 apps written for the iPhone and iPod touch over the past year.²
November, more than 100,000 apps and more than 2 billion downloads.

Apple provides Apple Developer Connection members a host of guidelines, tools, and documentation to help them get started³. The following is a list of some of the available resources.

- Getting Started Documents (*...tools, frameworks, development best-practices, and design methods for creating iPhone applications.*)
- iPhone Reference Library (*Explore a collection of in-depth technical documentation, sample code, guides, and articles for iPhone development categorized by topic and frameworks.*)
- iPhone Application Programming Guide
- iPhone Development Guide
- iPhone Human Interface Guidelines
- Your First iPhone Application
- Learning Objective-C: A Primer
- iPhone SDK 3.1.2
- Getting Started Videos
- Coding How-To's
- Sample Code
- iPhone Developer Program
- App Store Resource Center (*Find details on everything from how to prepare for submitting an app to managing an app once it's been posted.*)
- Join the iPhone Developer Program (*The iPhone Developer Program offers a complete process for developing and distributing iPhone or iPod touch applications.*)
- Developer Tools and Technologies (*Read about the tools and technologies that make developing applications for Mac and iPhone simple, yet incredibly powerful.*)
- iPhone Developers Tell Their Story (*Watch developers share how iPhone changed what they thought was possible on a mobile platform.*)
- iPhone OS Accessories (*Design applications to communicate with accessories connected through the 30-pin dock or using Bluetooth.*)
- WWDC Session Videos (*Session videos from the Worldwide Developers Conference 2009 are now available to purchase.*)

In addition to the above, Apple has teamed with Stanford to provide free videos and course materials on iPhone application Development.⁴

Critical Success Factors / Key Components

What are some of the key components that are making Apple's iPhone application development model so successful? Publishers? Distribution partners? Specialist development hardware? None of it is necessary. Below is a list of some of the more obvious critical success factors.

² Happy first birthday, App Store; July 13, 2009;

<http://iphoneappsdevelopment.wordpress.com/2009/07/13/happy-first-birthday-app-store/>

³ <http://developer.apple.com/iphone/>

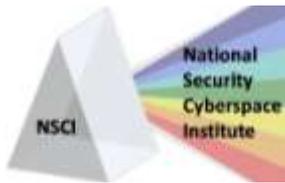
⁴ Apple, Standard Teaching iPhone Development for Free; April 2, 2009;

<http://www.wired.com/gadgetlab/2009/04/apple-stanford/>

Keeping Cyberspace Professionals Informed

12 Dec 2009

Larry McKee--Larry.McKee@nsci-va.org



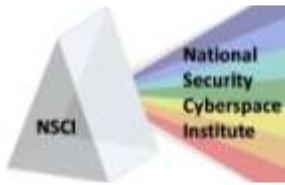
Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

- Easy to download and install (both the iPhone applications and the developer "tools")
- Applications are high on performance (e.g. app cannot more than 4.5mb data transfer in 5 minutes due to battery life)
- Popularity of iPhone (e.g. business and social aspects regarding iPhone)
- Global market of enthusiastic downloaders
- High number of iPhone users (thousands of potential buyers; enterprise to individual users) make .99 applications viable through sales quantity
- Large number of developers (*iPhone development companies being formed, traditional companies standing up iPhone development units, and independent developers*)
- Low development costs compared to application development for same projects at mainland development facilities
- Minimum cost of entry (e.g. Intel-based Mac running OS X 10.5.3 or later)
- Minimum learning curve (e.g. Apple's object oriented Objective-C framework - Cocoa - similar to other object oriented languages such as C++)
- Free development tools (e.g. xCode, Interface Builder, iPhone Simulator)
- Focused (e.g. small, simple) functionality in applications; but broad range of application potential (e.g. games, calendar, to-do lists, email, text messaging, social networking, maps/directions, location aware apps to take advantage of what's around you)
- The majority of applications do not require / use a database
- Open, free information has enabled universities to teach courses on iPhone development and publishers to publish books (e.g. iPhone Application Development for Dummies)
- Widespread publicity of iPhone App Store making some independent developers rich. For example, independent developer Steve Demeter [announced earning \\$250,000](#) in just two months with his game *Trism* (\$4.99). And Ethan Nicholas, developer of the iPhone game *iShoot* (\$2.99), raked in \$600,000 in a single month with sales of his app.
- Perspective that iPhone App Store "pries control of mobile device apps from the big-bad telco giants and puts it in the hands of average consumers and developers"⁵
- Free lancer iPhone developers beginning to emerge - for a fee, they will write code implementing your idea.
- Simple process / steps for burgeoning app stars...
 1. Buy a Mac
 2. Down the SDK (software development kit)
 3. Learn Objective C
 4. Start writing something!
 5. Sign-up as an official developer (\$99; agree to Apple's terms and conditions - including 30% commission)
 6. Prepare for a few weeks of work...
 7. Submit your app to Apple
 8. Adapt, market and survive!

Adaptation / Adoption Considerations, Requirements and Recommendations

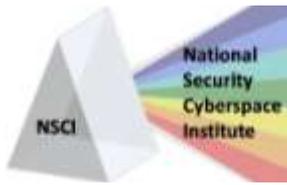
- User / customer pool is largely set - C2 Centers. Not a huge volume in these, so 0.99 not likely to work. Could increase user base by including admin/business apps (e.g. NIPRNET apps such as tasker manager, RFI manager, etc).

⁵ iPhone App Development - Where to Start; Dale Dietrich; September 12, 2008;
<http://www.daleisphere.com/iphone-app-development-where-to-start/>



Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

- One of the primary goals should be to get as large a potential developer pool as possible. Should include small business as well as large business.
 - Must you have a security clearance to do development?
 - Do you want to enlist global pool of developers, or limit to US citizens?
- Keep cost of entry to a minimum. Consider making as much of the development environment as possible "Government Furnished Equipment" (GFE) (e.g. cloud computing applied to development).
- Is the government able to confine / describe the target infrastructure (e.g. Windows Vista, Oracle database version, bandwidth limitations, programming language such as Java or C++)?
- Is the target infrastructure common enough to enlist a large pool of developers without them having to learn something new (e.g. leverage what they have already learned and what .edu already teaches - Java, C++, etc)
- Certification & Accreditation process for developers to get it on the network?
 - If you make it the responsibility of the developer, you will probably only get the traditional developers. If you can somehow make C&A independent of the developer, you will get larger pool of developers. Just need to publish C&A expectations/guidelines so developers can build to them; then govt do the testing before app is approved. Must be a fairly quick process or developers will lose interest - they don't want to develop an app this year and not get paid until next year.
- Can the government make required data "GFE" (government furnished equipment)? (e.g. provide antivirus software logs to those wanting to build cyber situational awareness app)
 - Unclassified data sets?
 - If classified data, will it need to be on-site development?
 - Large organizations may have closed secret network / development; but many small businesses do not. Very few contractors have SIPR at their location anymore.
- Applications involving the manipulation of data likely to involve the most risk. Applications requiring the use of data less risky. Are there application needs without data?
- Can the government break JC2 down into pieces that are small enough to allow industry to accept some of the risk?
 - Developers / designers cost somewhere between \$150-200 per hour (includes more than salary - all costs to get it to market). For a 3-man month project, that's about \$80k. Perhaps reasonable. But for a bigger project, something that takes 6 or 9 man months, that's \$150k - \$225k. Perhaps too much risk to take on alone.
 - At least initially, smaller apps likely to get more interest.
- Small business may view this as a way to more economically compete with large business; but will require JC2 capabilities to be broken down into reasonable pieces (e.g. not develop GCCS or TBMCS all at once, but develop in smaller capability modules that can be done in weeks to months).
- Can the government come up with a model to share risk with developers (e.g. matching "grant" on front-end to share some costs/risks; venture capital)
- Given a smaller potential pool of users, the 0.99 model will not work. What is a reasonable price?
 - Does the government purchase the software, then available to any user?
 - Does the user purchase the software?
 - 3600 vs 3400 funding limitations?
- Developers will need to provide ability to have a 5-7 day trial version so users can test run the app before committing their money to it.
- Anticipate developers will provide core functionality; don't anticipate lots of bells and whistles. This could be good...limit the complexity of the code - ease of use and less opportunity for security breach.



Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

- Consider periodic "developer conferences" to spread the word, recognize the big winners, share lessons learned / best practices, etc.
- Frequent press releases to encourage more developers to participate.

Lessons Learned

Apple failed to anticipate the volume of interested developers. Initially, requests for an agreement took as little as two days - very encouraging to developers. As developer interest increased, coders experienced delays that stretched to months of waiting for Apple to process their request and applications (both free apps and commercial apps).⁶ This created much frustration with App Store approval process.

Barriers to entry still keep some developers from jumping on. Only one language available (Objective C), one development platform (Mac), and one database (SQLite).

Niche apps do not sell enough to make .99 worthwhile; yet with that being the standard - many users are less willing to purchase more expensive apps (especially those above \$4.99).

Apps with robust functionality require more time and developers, thus .99 not a reasonable price unless very large number of users.

Free and .99 = limited lifespan and broad appeal (e.g. ringtones).

Of note, there are some who do not embrace the iPhone application development model. Many feel the Apple review process sets a horrible precedent and that other software platforms may soon implement similar "gatekeeper" processes.⁷ Their belief is based on the internet being free, and apps should not have to go through a gatekeeper to get to users.

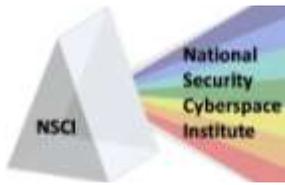
⁶ Would-be iPhone developers "pulling their hair out by the roots"; March 7, 2009;
http://www.appleinsider.com/articles/09/03/07/would_be_iphone_developers_pulling_their_hair_out_by_the_roots.html

⁷ Facebook iPhone Dev Quits Project Over Apple Tyranny; Jason Kincaid; November 11, 2009;
<http://www.techcrunch.com/2009/11/11/joe-hewitt-developer-of-facebooks-massively-popular-iphone-app-quits-the-project/>

Keeping Cyberspace Professionals Informed

12 Dec 2009

Larry McKee--Larry.McKee@nsci-va.org



Apple iPhone Application Development Model: A Possible Approach for JC2 Application Development?

What are typical 'gotcha' during the iPhone application approval process?

1. **Violation of Apple's Human Interface Guidelines (HIG)** is probably the most frequent cause for an app to be rejected.
2. **Matching icons** Believe it or not, Apple is now requiring the 512x512 iTunes Store icon match the 57x57 icon displayed on the iPhone.
3. **Simulating failures** Apple doesn't like anything that pretends the iPhone or iPod Touch is failing.
4. **Button images must be consistent** If you decide to use one of the existing images Apple provides for buttons, be careful you use it for an identical function.
5. **Bandwidth usage over cellular networks** If your app downloads data over the cellular network, ensure you do not use too much bandwidth. How much is too much? Well, there isn't an exact number, but a tech support person from Apple advised me to not exceed 4.5 meg of data per 5 minutes of activity.
6. **Popup for network detection** If your app requires the use of the Internet, you must detect when the network is unavailable and provide a pop-up message informing the user.
7. **False claims of a missing network** On a related note, make sure you don't have any false positives in your network detection.
8. **Political lampooning** Don't make any jokes about political figures, past or present, in either your app or the description in iTunes.
9. **OS compatibility** If you claim your app works with OS 2.0 and higher, you better make sure you test whether your app really does work on all the OS versions between 2.0 and the current one.
10. **Trademarks, Particularly Icons** — Numerous apps ran into delays and rejections for including icons and imagery that a Apple deemed a trademark violation. Common culprits: iPhone-like icons and Polaroid-like image frames.
11. **Giveaways/Prize Apps/Contests** —Apple rejects prize applications and apps that contains contests or giveaways. There are exceptions to this policy.
12. **Don't Ask, Don't Tell** — ...package used in thousands of apps that collects anonymized usage data. As a courtesy to his users, Alan stated that he was collecting this data *and provided an opt-out mechanism*. EHis app was rejected until he added a giant warning label on first run.
13. **Inadvertent "Objectionable Content"**
14. **Update Spam** — There's some indication that Apple frowns upon publishing no-change updates in an attempt to keep your app appearing in the what's new listings.
15. **Public Figures** — Brian's original article included "political lampooning." I'll extend that to include association or portrayal of public figures.
16. **Too Few Potential Consumers (Or The Appearance Thereof)** —Best bet is to save them the work by supplying them with evidence in your submission has a vast, mainstream audience — or at least a sizable niche one.
17. **Don't Include Price In Your Description** — "Don't mention pricing in the App Description. For example mentioning 'now only \$1.99' will according to Apple, 'potentially confuse users'...and they have a point as its 99 pence in the UK, €1.99 in Europe etc."

iPhone Application "Categories"

Apps for Cooks; Apps for Keeping Current; Apps for the Great Outdoors; Apps for Music; Apps for Work; Apps for Students; Apps for Moms and Dads; Apps for Working Out; Apps for Going Out; Apps for Managing Money; Apps for Traveling; Apps for Fun and Games