

**CASE STUDY AND REPORT****Available Online at www.ijarcs.info****History and Features of a developer tool in iOS: XCODE**

Miss Priyanka V. Kanoi
2nd yr Information and Technology
JDIET , Yavatmal
kanoi.priyanka@gmail.com

Miss Payal N. ingole
3rd yr Computer sc and Engineering
JDIET, Yavatmal
ingole.payal@gmail.com

Abstract: The powerful developer tool Xcode developed by apple.inc provides everything for creating great apps in mac, iphone and ipad. XCODE is integrated tightly with the Cocoa and Cocoa Touch frameworks. XCODE creates powerful and easy to use development environment, which is powerful enough to be used by OS X and iOS. Xcode toolset includes Xcode IDE, with the Interface Builder design tool and fully integrated LLVM compiler. Dozens of other supporting developer tools are also included in XCODE along with Instruments analysis tool.

Keywords: XCODE, XCODE IDE, LLVM compiler, Simulators, LLDB Debugger.

I. INTRODUCTION

Xcode is an Integrated Development Environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. First released in 2003, the latest stable release is version 4.6. Registered developers can download preview releases and previous versions of the suite through the Apple Developer website.[1]

The main application of the suite is the integrated development environment (IDE), also named Xcode. The Xcode suite also includes most of Apple's developer documentation, and built-in Interface Builder, an application used to construct graphical user interfaces. The Xcode suite includes a modified version of the GNU Compiler Collection as well as, in Xcode 3.1 and later, the LLVM-GCC compiler, with front ends from the GNU Compiler Collection and a code generator based on LLVM, and, in Xcode 3.2 and later, Apple's LLVM Compiler,[2] with the clang front end and a code generator based on LLVM, and the Clang Static Analyzer. It supports C, C++, Objective-C, Objective-C++, Java, AppleScript, Python and Ruby source code with a variety of programming models, including but not limited to Cocoa, Carbon, and Java. Third parties have added support for GNU Pascal, Free Pascal, Ada, C#, Perl, and D. The Xcode suite used the GNU Debugger as the back-end for its debugger. As of version 4.2 the Apple LLVM Compiler became the default compiler.[3] LLDB became the default debugger as of Xcode 4.3.

II. VERSION HISTORY AND ANALYSIS**A. X 1 series:**

Xcode 1.0 was released in fall 2003. Xcode 1.0 was based on Project Builder, but had an updated UI, ZeroLink, Fix & Continue, distributed build support, and Code Sense indexing. The next significant release, Xcode 1.5, had better code completion and an improved debugger.

B. X 2 series:

Xcode 2.0 was released with Mac OS X v10.4 "Tiger". It included the Quartz Composer visual programming language, better Code Sense indexing for Java, and Ant support. It also included the Apple Reference Library tool. Xcode 2.1 could create universal binaries. It supported Shared Precompiled Headers, unit testing targets, conditional breakpoints, and watchpoints. It also had better dependency analysis. The final version of Xcode for Mac OS X v10.4 was 2.5.

C. X 3 Series:

Xcode 3.0 was released with Mac OS X v10.5 "Leopard". Notable changes since 2.1 include [4] the DTrace debugging tool (now called Instruments), refactoring support, context-sensitive documentation, and Objective-C 2.0 with garbage collection. It also supports Project Snapshots, which provide a basic form of version control; Message Bubbles, which show build errors debug values alongside code; and building four-architecture fat binaries (32 and 64-bit Intel and PowerPC).

Xcode 3.1 was an update release of the developer tools for Mac OS X, and was the same version included with the iPhone SDK. It could target non-Mac OS X platforms, including iPhone OS 2.0. It included the GCC 4.2 and LLVM GCC 4.2 compilers. Another new feature since Xcode 3.0 is that Xcode's SCM support now supports Subversion 1.5. Xcode 3.2 was released with Mac OS X v10.6 "Snow Leopard" and will not install on any earlier version of Mac OS X. It supports static program analysis, among other features. It also drops official support for targeting versions earlier than iPhone OS 3.0. But it is still possible to target older versions, and the simulator supports iPhone OS 2.0 through 3.1. Also, Java support is "exiled" in 3.2 to the organizer. [5] Xcode 3.2.6 is the last version that can be downloaded for free for users of Mac OS X v10.6.

D. X 4 series:

In June 2010 at the World Wide Developers Conference, Apple announced version 4 of Xcode during the Developer

Tools State of the Union address. Version 4 of the developer tools consolidates the Xcode editing tools and Interface Builder into a single application, among other enhancements.[6][7] Apple released the final code for Xcode 4.0 on March 9, 2011. Xcode 4.0 drops support for many older systems, including all PowerPC development as well as SDKs for Mac OS X 10.4 and 10.5, and all iOS SDKs older than 4.3.

The deployment target can still be set to produce binaries for those older platforms, but for the Mac OS platforms one is then limited to creating x86 and x86_64 binaries.

Xcode 4.1 was made available on July 20, 2011. Xcode 4.1 was the last version to include GCC instead of only LLVM GCC.

On October 12, 2011, Xcode 4.2 was released concurrently with the release of iOS 5.0, and it included many more and improved features, such as storyboarding and automatic reference counting (ARC). Xcode 4.2 is the last version to support Mac OS X 10.6 "Snow Leopard".

Xcode 4.3, released on February 16, 2012, is distributed as a single application bundle, Xcode.app, installed from the Mac App Store. Xcode 4.3 reorganizes the Xcode menu to include development tools. Xcode 4.3.1 was released on March 7, 2012 to add support for iOS 5.1. Xcode 4.3.2 was released on March 22, 2012 with enhancements to the iOS Simulator and a suggested move to the LLDB debugger as opposed to the GDB debugger (which appear to be un-documented changes). Xcode 4.3.3, released in May 2012, featured an updated SDK for Mac OS X 10.7.4 "Lion" and a few bug fixes.

Xcode 4.4 was released on July 25, 2012. It runs on both Mac OS X Lion (10.7) and OS X Mountain Lion (10.8) and is the first version of Xcode to contain the OS X 10.8 "Mountain Lion" SDK. Xcode 4.4 includes support for automatic synthesizing of declared properties, new Objective-C features such as literal syntax and subscripting, improved localization. On August 7, 2012, Xcode 4.4.1 was released with a few bug fixes.

Xcode 4.5 was released on September 19, 2012, on the same day that iOS 6 was released. It added support for iOS 6 and the 4-inch Retina display found on iPhone 5 and iPod touch 5th generation. It also brought some new Objective-C features to iOS, simplified localization, and added auto-layout support for iOS.[8] On October 3, 2012, Xcode 4.5.1 was released with bug fixes and stability improvements. Less than a month later, Xcode 4.5.2 was released, with support for iPad mini and iPad with Retina display, and bug fixes and stability improvements.

Xcode 4.6 was released on January 28, 2013, on the same day that iOS 6.1 was released.

III. MAJOR FEATURES OF XCODE

Previously Xcode supported distributing a product build process over multiple systems. One technology involved was

called Shared Workgroup Build, it used the Bonjour protocol to automatically discover systems providing compiler services, and a modified version of the free software product DISTCC to facilitate the distribution of workloads. Earlier versions of Xcode provided a system called Dedicated Network Builds. These features have been absent in the supported versions of Xcode.

Xcode can build universal binaries which allow software to run on both PowerPC and Intel-based (x86) platforms and that can include both 32-bit and 64-bit code for both architectures. Using the iOS SDK, Xcode can also be used to compile and debug applications for iOS that run on the ARM processor.

Xcode also includes Apple's WebObjects tools and frameworks for building Java web applications and web services (previously sold as a separate product). As of Xcode 3.0, Apple dropped WebObjects development inside Xcode; WOLips [9] should be used instead. Xcode 3 still includes the WebObjects frameworks.

Xcode includes the GUI tool Instruments, which runs atop DTrace, a dynamic tracing framework created by Sun Microsystems and released as part of Open Solaris.

IV. THE XCODE TOOLSET

A. Xcode IDE:

Xcode integrates all the tools you need. The unified interface smoothly transitions from composing source code, to debugging, and even to designing your next user interface, all within the same window. The Xcode workspace is all about keeping you focused.[10] As you type, Live Issues will immediately alert you to coding mistakes, displaying a message bubble beside your code for more detail. Hit the Run button to launch your Mac app, or upload the app to your test device, and immediately start debugging. Hover the mouse pointer above a variable to inspect its value at runtime, never having lost your place in the editor. By working closely with the developer web portal, Xcode provisions new iOS devices with a single click, can securely sign and archive your Mac or iOS app, and directly submit it to the App Store.

Other features included in XCODE IDE are

- a. Assistant
- b. version editor
- c. interface builder build-in
- d. integrated build system
- e. static analysis
- f. Organizer
- g. Snapshots
- h. Refactoring
- i. complete documentation
- j. quick help
- k. version editor
- l. live issues
- m. fix it

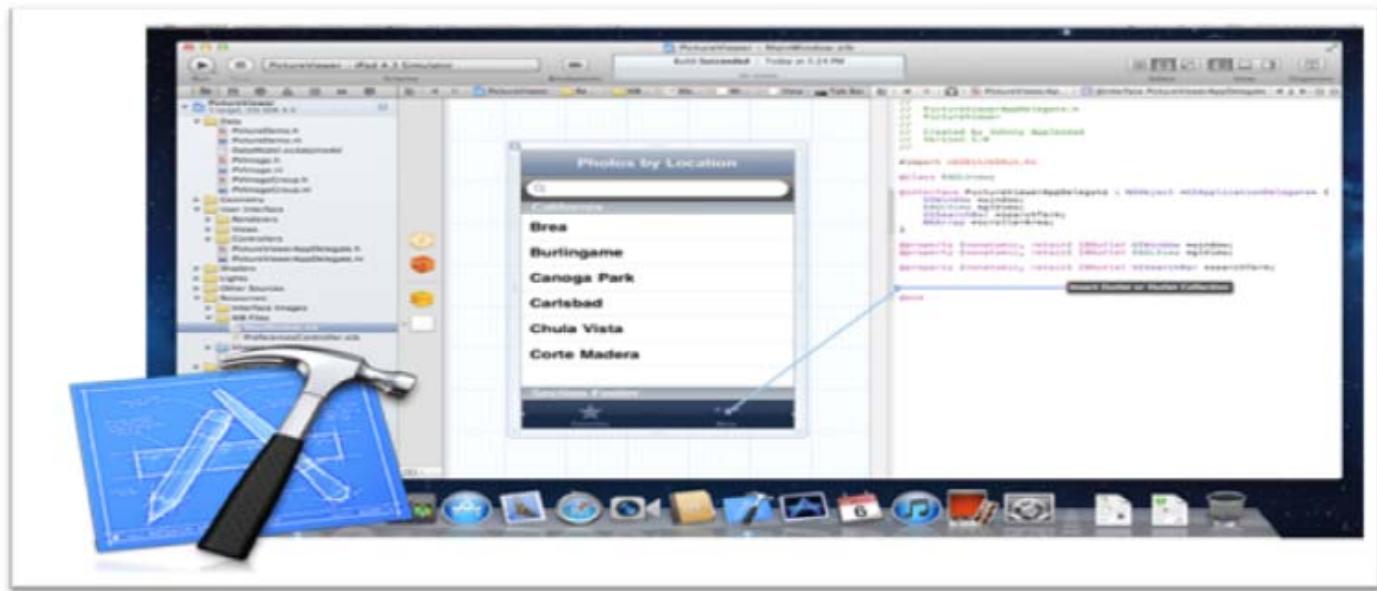


Figure: 1

B. LLVM Compiler:

LLVM compiler does more than build your app. LLVM technology is integrated into the entire development experience. The same parser used to build C/C++ and Objective-C powers Xcode's indexing engine, providing incredibly accurate code completions. As you work, LLVM is constantly evaluating what you type, identifying coding mistakes that Xcode shows as Live Issues, and thinking ahead for ways to Fix-it for you. Other compilers can tell you what is wrong —Xcode LLVM can make it right.

C. Instruments for Performance and Behavior Analysis:

The Xcode developer tools include Instruments, unique application that helps you track down performance bottlenecks in your OS X and iOS apps.

Instruments collect data such as disk, memory, or CPU usage in real time, either on your Mac or remotely from a connected iPhone. The collected data is graphically displayed as tracks over time, making it easy to pinpoint problem areas, then drill down to the offending lines of code. Instruments helps guarantee your user interface will be accompanied by equally impressive responsiveness.



Figure: 2

D. iOS Simulator:

The iOS Simulator runs your application in much the same way as an actual iOS device. Because it is quick to launch and debug, the iOS Simulator makes for a perfect test bed to make sure your user interface works the way you intend, your network calls are correct, and that the views change correctly when the phone rotates. You can even simulate touch gestures by using the mouse.

V. XCODE 4

With Xcode 4, the tools have been redesigned to be faster, easier to use, and more helpful. The Xcode IDE understands your project's every detail, identifies mistakes in both syntax and logic, and will even fix your code for you. Xcode 4 has a brand new user interface, built upon proven technologies that Apple itself uses to build OS X and iOS.

VI. FEATURES OF XCODE 4

A. Single Window:

The first thing you will notice about Xcode 4 is that the many windows used to perform the development tasks you work on every day have been consolidated into a single window. The Xcode 4 work area has several unique UI elements that make it easy to work on many different tasks, even multiple projects, without cluttering your work area.

B. Navigators:

On the left side is a collection of navigators, including the list of files in your project, sorted symbols, a central search interface, issue tracking, debugging data with compressible stack traces, active and inactive breakpoints, and a persistent collection of logs. The unified navigator UI provides live filtering of content and search results.

C. Jump Bar:

At the top of every editor pane is a path bar showing the relative location of your current file. Click any location in the path to immediately jump to any other file at that level. This is the Jump Bar, and it is so efficient you may wish to dedicate your entire desktop to source code, quickly jumping from file to file.

D. Interface Builder is Built-in:

In Xcode 4, Interface Builder has been completely integrated within the Xcode IDE — there is no separate application. Selecting an interface file (.nib/.xib) in your project will open the IB editor within Xcode. Opening the right-hand Utility area will show the full complement of interface inspectors, as well as the library of controls and UI objects. Drag a control from the library, and drop onto the canvas, to layout your OS X or iOS app. The best part: you can drag connections directly from the UI design to the source code.

E. Assistant:

Whether writing code, or designing an interface, no change happens in isolation. Even the smallest edit can have a cascading effect throughout your project. You often need to see more than just your current file; you need to see related documents as well. With the new Xcode Assistant, the two-pane editor layout in Xcode 4 becomes dramatically more powerful. When you turn on Assistant, the IDE will anticipate which other files you need to see, as you work. The Assistant will show you the code for the class you are inheriting. The Assistant will automatically show you the corresponding header. When designing an interface, the Assistant will show you the appropriate controller, making drag-and-drop code connections extremely simple. Data model designing will bring up the classes that back your models — all automatically.

F. Instruments for XCODE:

Instruments have a streamlined interface, complete with the new Jump Bar, and stack compression from Xcode 4's UI.

Using Xcode4's new launch schemes, it is easier than ever to create a robust test harness to run your app in Instruments. That includes launching Instruments in "deferred mode", keeping the data collection UI off screen and saving system resources for running and metering your application. New data collection instruments are also available, including OpenGL ES for tracking iPhone graphics performance, new memory allocation monitoring that can find unintended memory growth, Time Profiler on iOS for collecting samples with very low overhead, and complete System Trace for insight into how all system processes interact. Instruments covers even more ground, has even lower overhead, and is more useful than ever before.

VII. CONCLUSION

You need an Intel-based Macintosh computer and the Xcode tools, to develop iOS apps. Xcode is Apple's suite of development tools that provide support for project management, code editing, building executables, source-level debugging, source-code repository management, performance tuning, and much more. At the center of this suite is the Xcode application itself, which provides the basic source-code development environment. Xcode has gone through many stages of advancements. Its latest version xcode 4 have many added features for simplifying the task of building apps.

VIII. ACKNOWLEDGMENT

We thank Prof. Dr. Rajkishor Tugnayat sir (H.O.D) I.T Dept JDIET,Yavatmal for inspiring and guiding us. We extend our thanks to Ankita Patil mam and Kinjal Patel mam (Assistant Prof. CSE Dept JDIET,Yavatmal) for helping us out. And we are also thankful to our senior Mr.Pratik Thakare encouragement.

IX. REFERENCES

- [1] "Mac Dev Center". Apple Developer. Retrieved 2011-07-31.
- [2] Prince McLean (2008-06-20). "Apple's other open secret: the LLVM Compiler". Retrieved 2012-04-03.
- [3] "What's New In Xcode: New Features in Xcode 4.2". Apple. Retrieved 2012-03-28
- [4] Apple - Mac OS X Leopard - Features - 300+ New Features
- [5] "Apple Developer Tools for Java". Apple. Retrieved 2011-02-08
- [6] "WWDC Software Notes: Safari 5, Xcode 4, iTunes 9.2". Mac Rumors. 2010-06-08. Retrieved 2010-06-21.
- [7] ^ "Developer Tools - What's New in Xcode 4". Apple Developer. Retrieved 2011-02-08.
- [8] "What's New in Xcode 4.5". Apple Inc.. Retrieved 20 September 2012
- [9] "WOLips". Wiki.objectstyle.org. 2010-02-27. Retrieved 2010-06-21.
- [10] <https://developer.apple.com/technologies/tools>