

Building libraries for iOS

Going native

Alexander Dodatko
2014

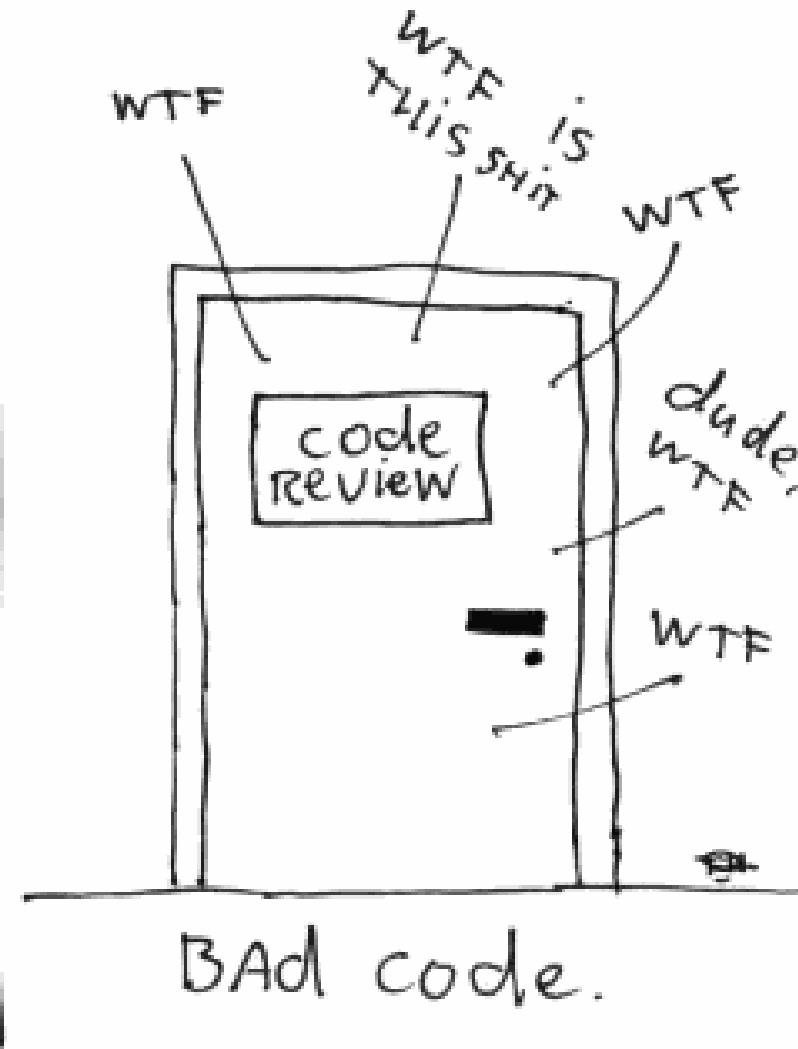
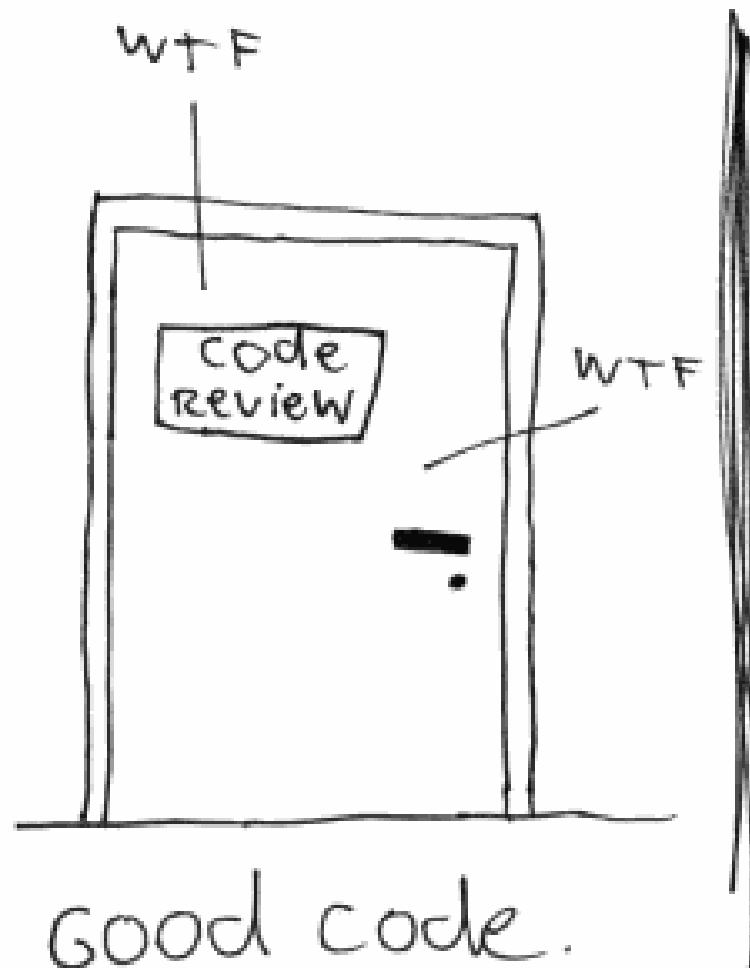


The standard library saves
programmers from having
to reinvent the wheel.

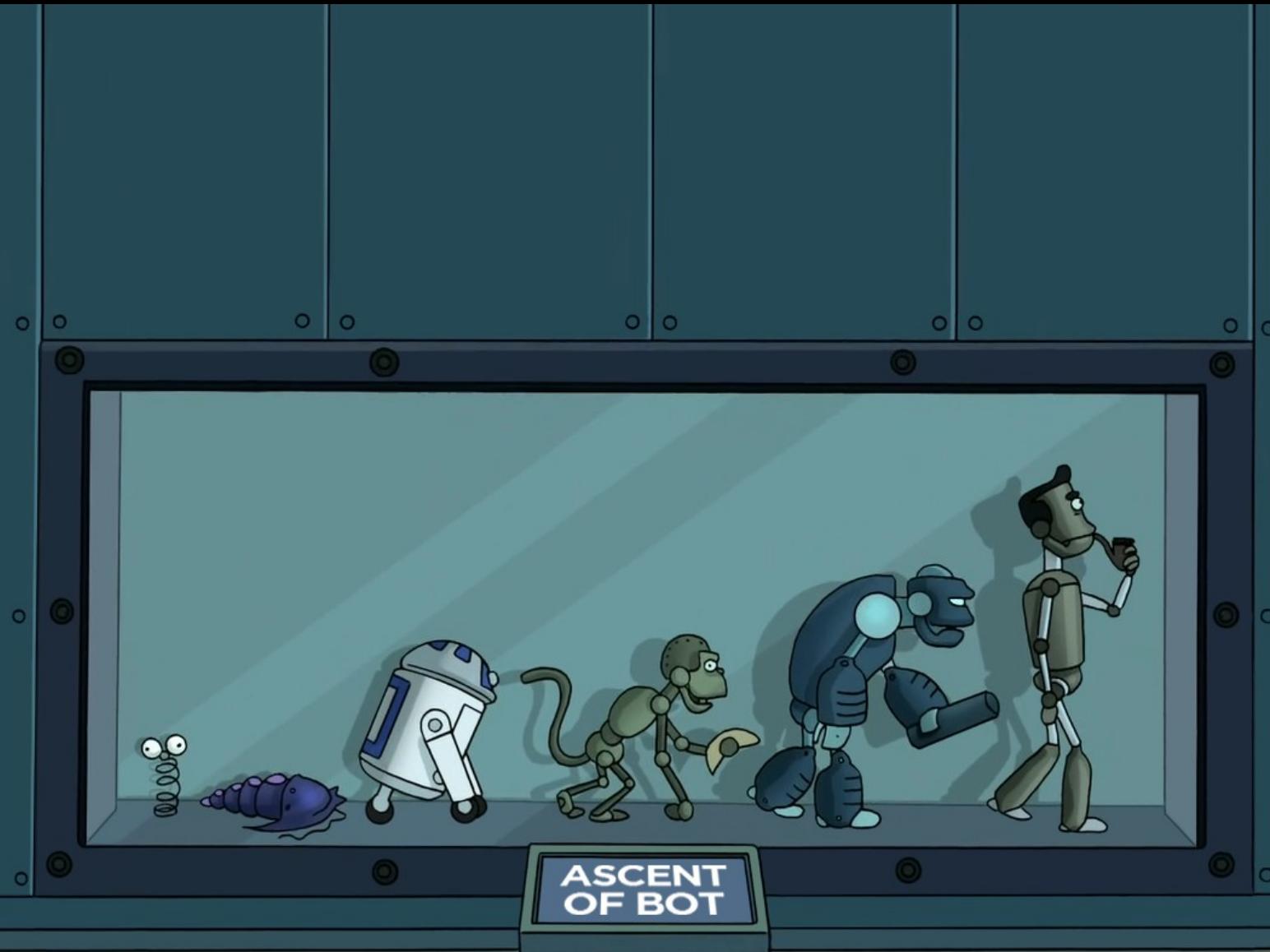
Bjarne Stroustrup

<http://bit.ly/1iqZ3JO>

The ONLY VALID MEASUREMENT OF Code QUALITY: WTFs/minute



Code Reuse is Important



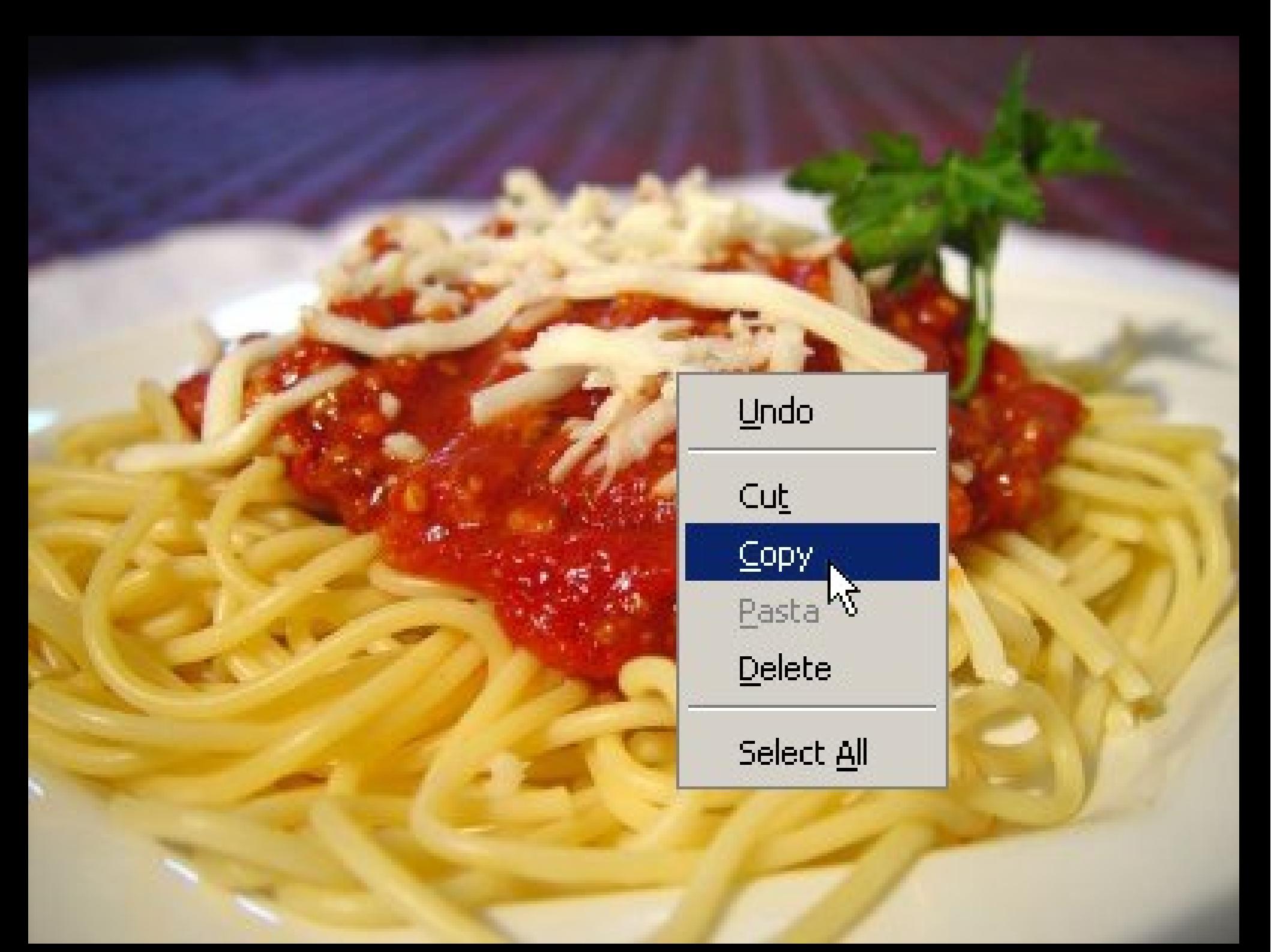
March 2008



<http://engt.co/1pOLKH9>

The fastest code is the code
that reaches the market first

Nobody Cares

A close-up photograph of a plate of spaghetti. The pasta is coated in a vibrant red tomato sauce and topped with melted white cheese. A small sprig of green parsley is visible in the background. A context menu is overlaid on the image, centered on a portion of the spaghetti. The menu has a light gray background with black text and a dark blue highlight. The options are: Undo, Cut, Copy, Paste, Delete, and Select All. The 'Copy' option is highlighted with a blue rectangle and a white cursor arrow is pointing to its right.

Undo

Cut

Copy

Paste

Delete

Select All

August 2011



CocoaPods
@CocoaPods



Follow

@oliverfoggin Thanks! Just over one year old:
github.com/CocoaPods/Coco...

Reply Retweet Favorite More



CocoaPods

An Objective-C library dependency manager. Contribute to CocoaPods development by creating an account on GitHub.

[View on web](#)



4:36 AM - 2 Oct 2012

Flag media



CocoaPods

pod install SomeAwesomeLibXYZ



Except...



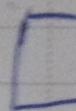




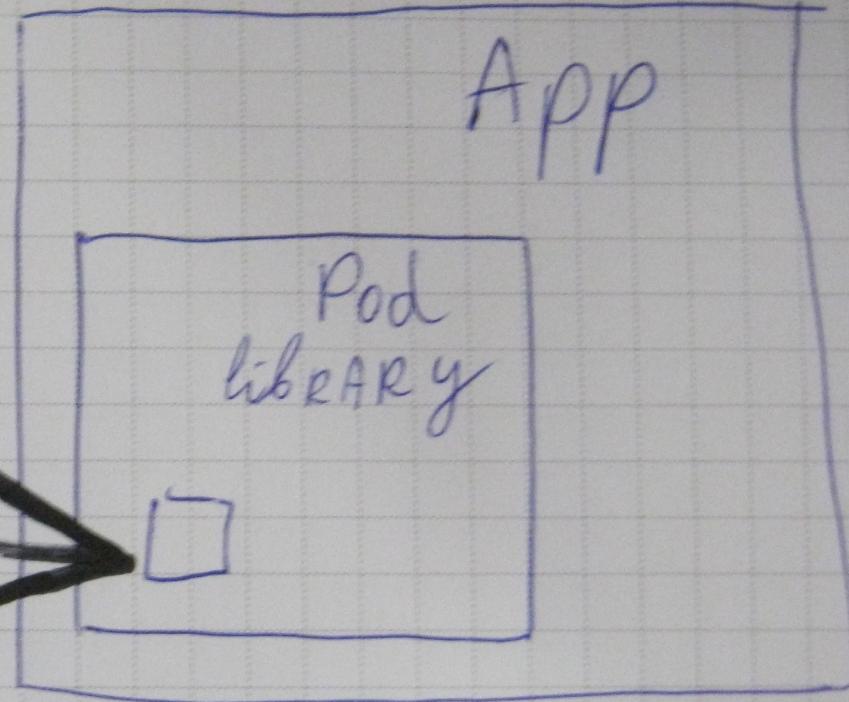
CocoaPods

Typical Workflow

Github



files
Pod install



Github

APP

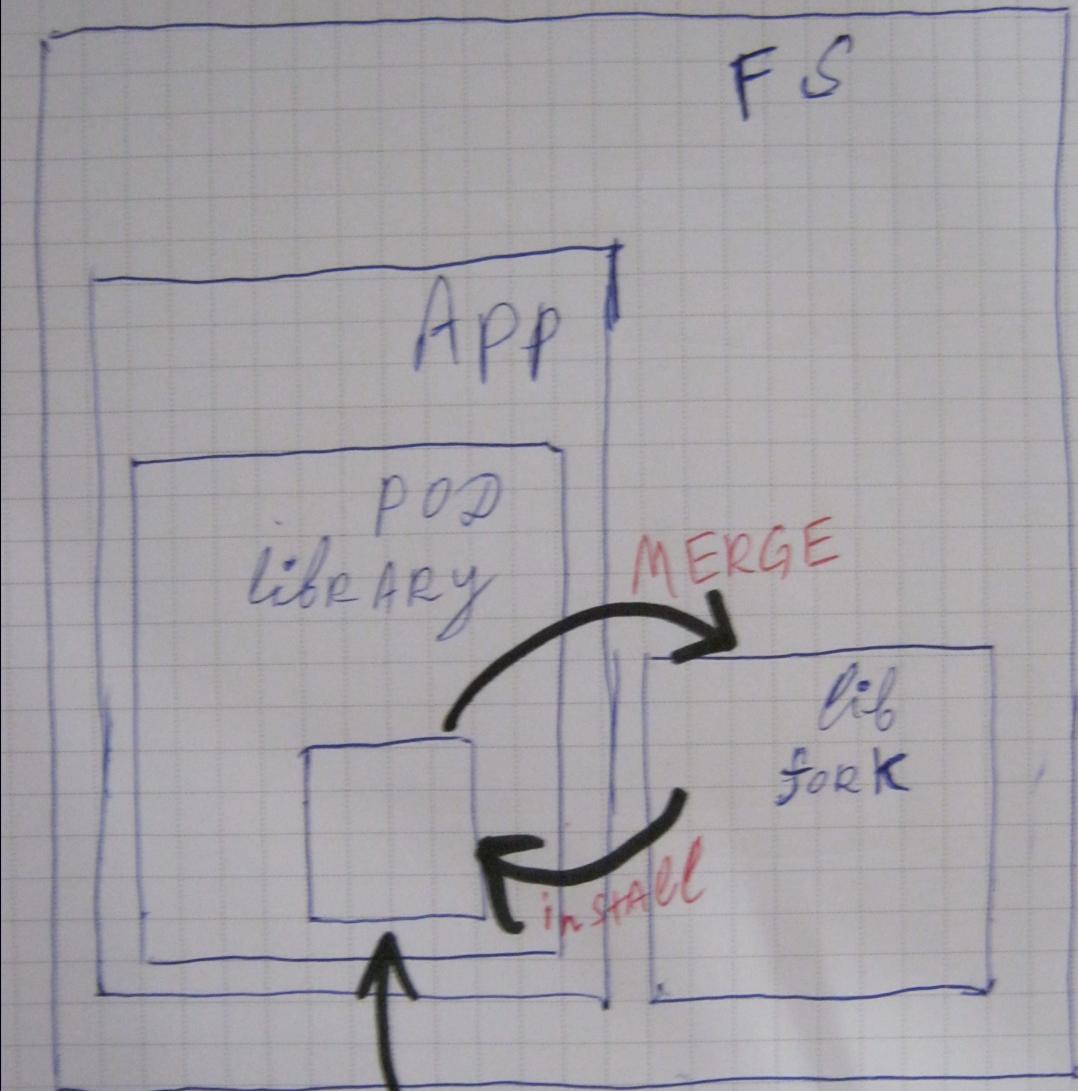
Siles
Pod Install

Pod
library



feature

F S



Debug
and
Fix

Our own Busyness Logic

App

.UI

Model

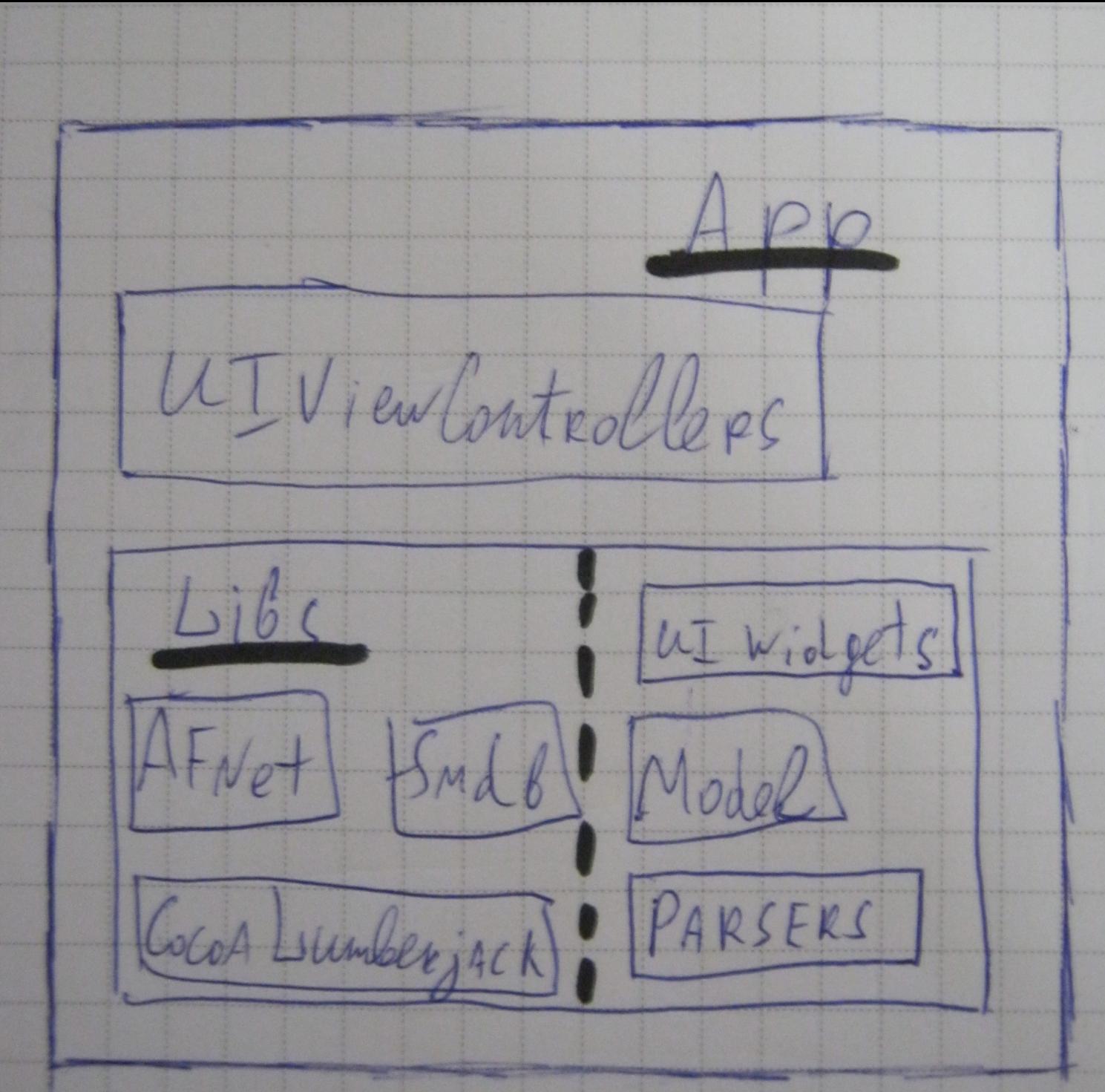
Libs

AFNet

fmdb

CocoALumberjack

In-Place Editing

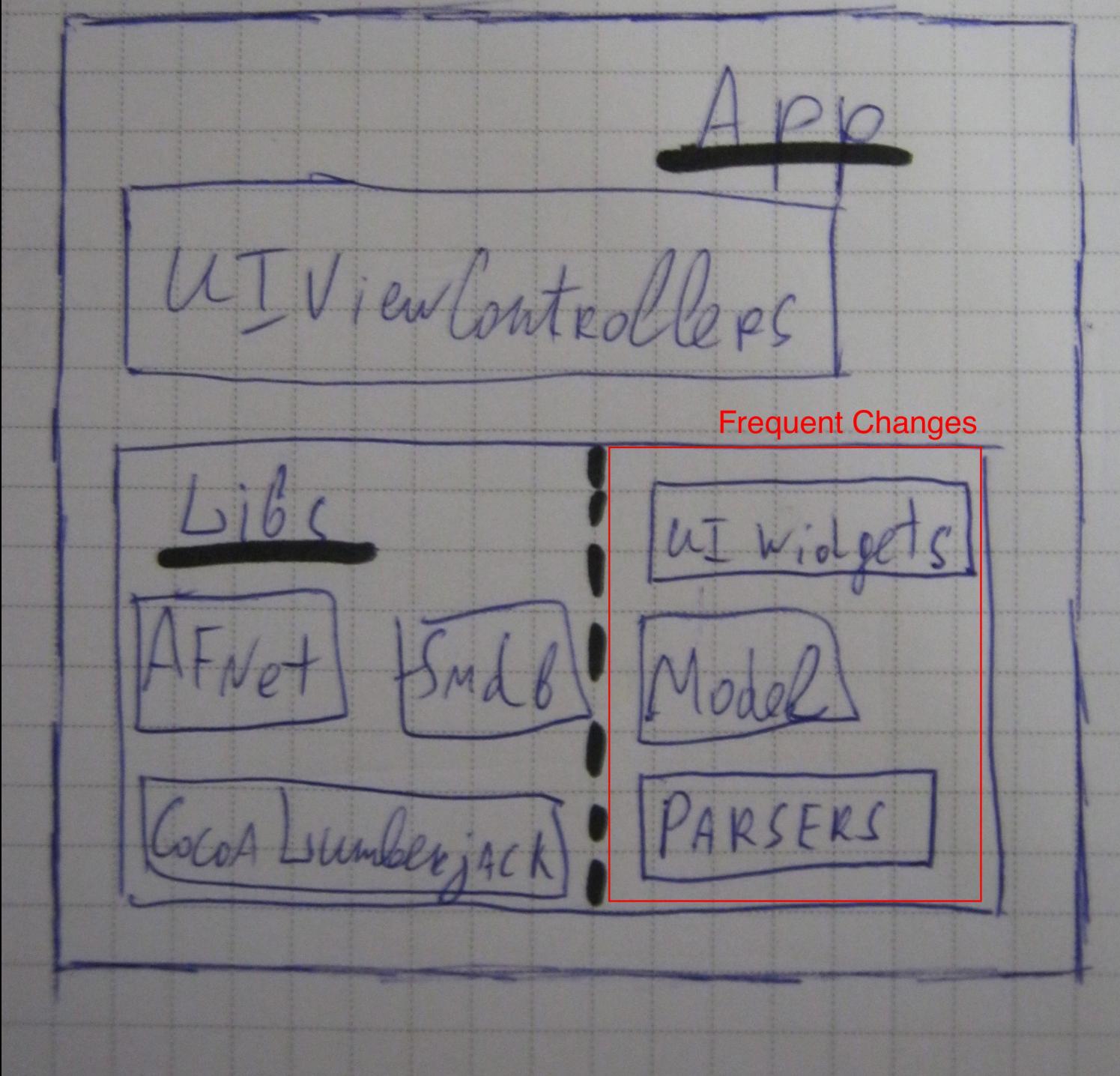


Modular Architecture Benefits

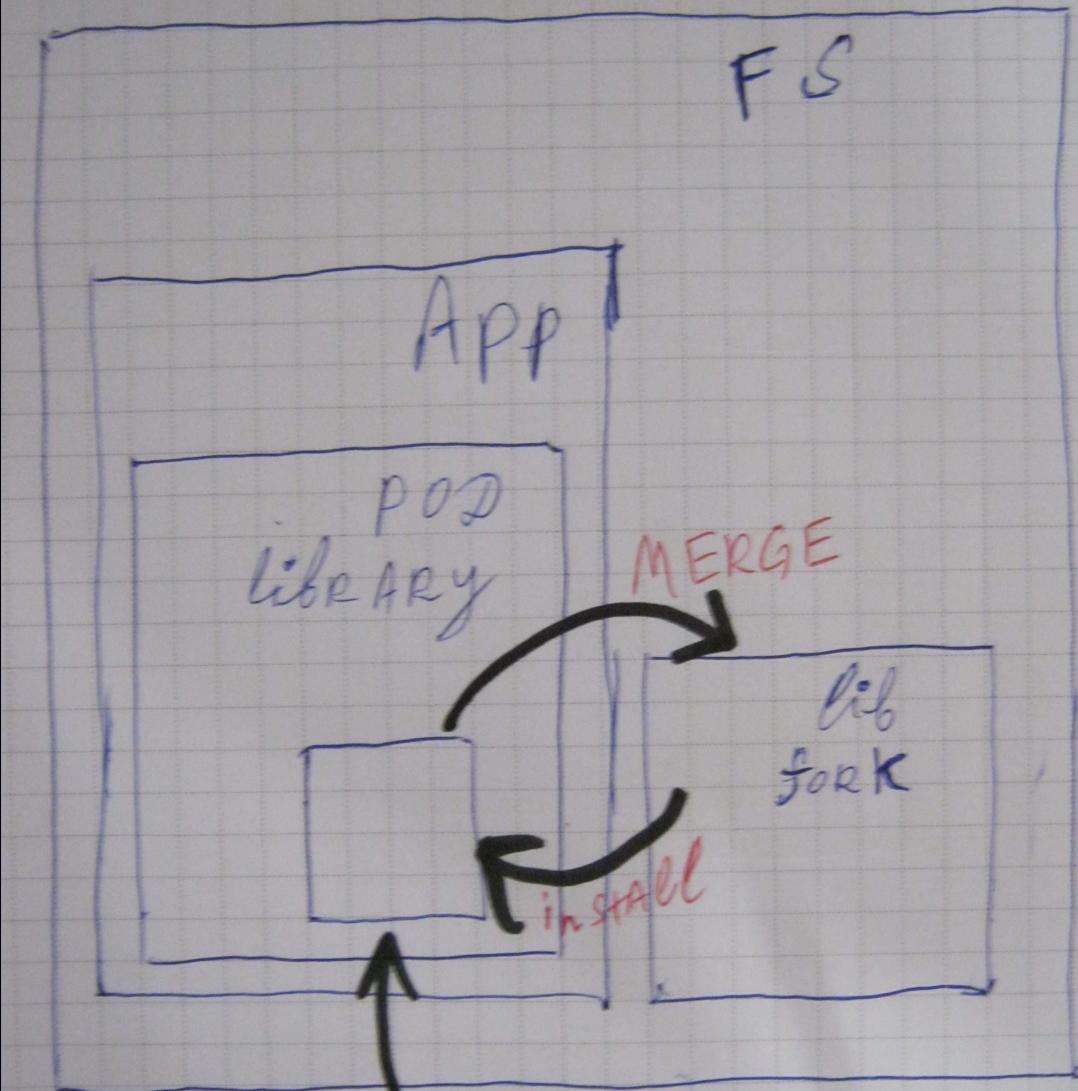
Code Reuse

Test Coverage

Easier to Apply Changes

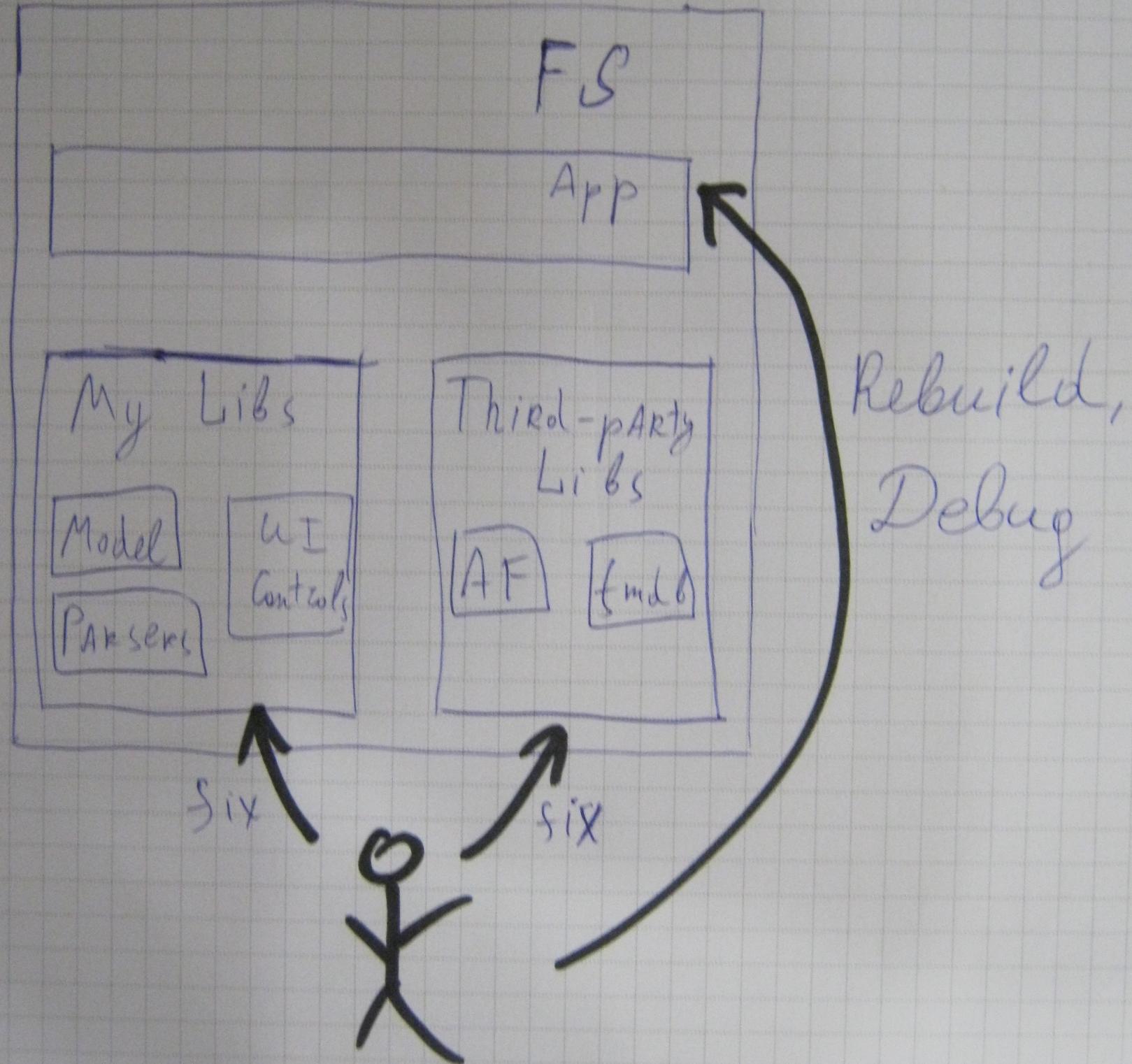


F S



Debug
and
Fix

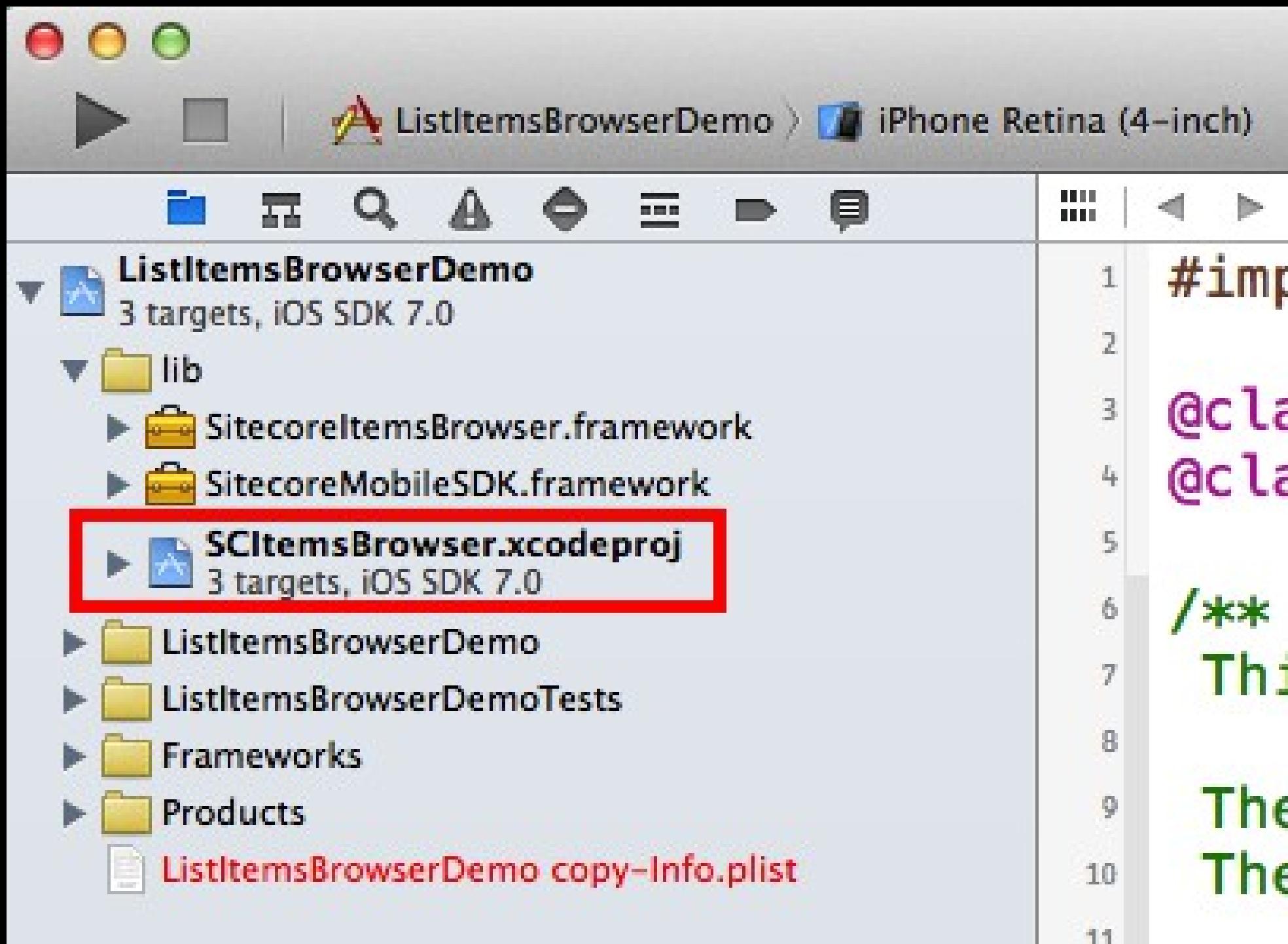
Going Native





CocoaPods

--no-integrate



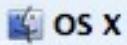
Choose a template for your new target



Application

Framework & Library

Other



Application

Framework & Library

Application Plug-in

System Plug-in

Other



Fake Static iOS
Framework



Static iOS
Framework



Cocoa Touch Static
Library

Cancel

Previous

Next

Not Just Library Target



#import <AFNetworking/AFNetworking.h>

VS

#import "AFHTTPRequestOperation.h"



Incapsulation



\$(inherited)

/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/... non-recursive

.

./frameworks non-recursive

./frameworks/SitecoreMobileSDK.framework/Headers recursive

+ -

A screenshot of a file search interface. The search results list includes the current directory, the Xcode toolchain path, and the ./frameworks directory. The ./frameworks entry is currently selected, highlighted with a blue background. A context menu is open over this selection, showing two options: "non-recursive" (which is checked) and "recursive". The entire context menu area is enclosed in a red rectangular border.

For Implementation Files

```
#import "AFHTTPRequestOperation.h"
```

▼ Search Paths

Setting	 ListItemsBrowserDemo
Always Search User Paths	No ▾
Framework Search Paths	.../../frameworks
Header Search Paths	/Applications/Xcode.app/Cont...
Library Search Paths	.../../frameworks
Rez Search Paths	
Sub-Directories to Exclude in Recursive Searches	*.nib *.lproj *.framework *.gch
Sub-Directories to Include in Recursive Searches	
User Header Search Paths	

For Public Headers

#import <AFNetworking/AFNetworking.h>

▼ Search Paths

Setting	
Always Search User Paths	No
Framework Search Paths	.../../frameworks
Header Search Paths	/Applications/Xcode.app/Cont...
Library Search Paths	.../../frameworks
Rez Search Paths	
Sub-Directories to Exclude in Recursive Searches	*.nib *.lproj *.framework *.gch
Sub-Directories to Include in Recursive Searches	
User Header Search Paths	

```
#import <AFNetworking/AFNetworking.h>
```



```
#import <Foundation/Foundation.h>
```

Static Framework for iOS



	libOCMock.a
▼	OCMock
	NSNotificationCenter+OCMAdditions.h
	OCMArg.h
	OCMConstraint.h
	OCMock.h
	OCMockObject.h
	OCMockRecorder.h



▼	OCMock.framework
	▼ Headers
	NSNotificationCenter+OCMAdditions.h
	OCMArg.h
	OCMConstraint.h
	OCMock.h
	OCMockObject.h
	OCMockRecorder.h
	OCMock
▼	Resources
	Info.plist



Drag & Drop



Framework Search Path is updated by Xcode



```
<plist version="1.0">
<dict>
    <key>CFBundleDevelopmentRegion</key>
    <string>English</string>
    <key>CFBundleIdentifier</key>
    <string>org.ocmock</string>
    <key>CFBundleInfoDictionaryVersion</key>
    <string>6.0</string>
    <key>CFBundlePackageType</key>
    <string>FMWK</string>
    <key>CFBundleSignature</key>
    <string>????</string>
    <key>CFBundleVersion</key>
    <string>2.2.3</string>
</dict>
</plist>
```



Frameworks Reduce Compilation Time



CocoaPods

CocoaPods Makes you Care about Versioning



For Development
and Unit Testing



For Distribution



For You



For End Users

Library Dependencies and Linker



```
ld: duplicate symbol _main in /Users/samuelleeuwenburg/Library/Developer/Xcode/D  
SDLStackOverflowTest1-awqtenipezizyreqnvxupfbheywd/Build/Intermediates/SDLStack  
Debug/SDLStackOverflowTest1.build/Objects-normal/x86_64/SDLMain.o and /Users/sa  
Library/Developer/Xcode/DerivedData/SDLStackOverflowTest1-awqtenipezizyreqnvxup  
Intermediates/SDLStackOverflowTest1.build/Debug/SDLStackOverflowTest1.build/Obj  
main.o for architecture x86_64  
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

Duplicate symbol _main in /Users/samuelleeuwenburg/Library/Developer/Xcode/DerivedData/SDLStackO

! Linker command failed with exit code 1 (use -v to see invocation)



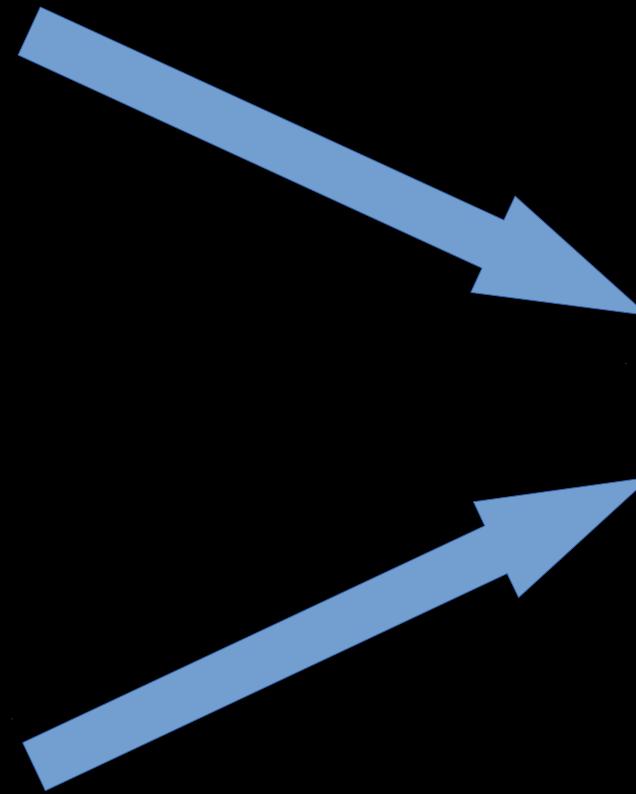
Activity Log Complete 9/26/12 3:41 PM

1 error, 1 warning



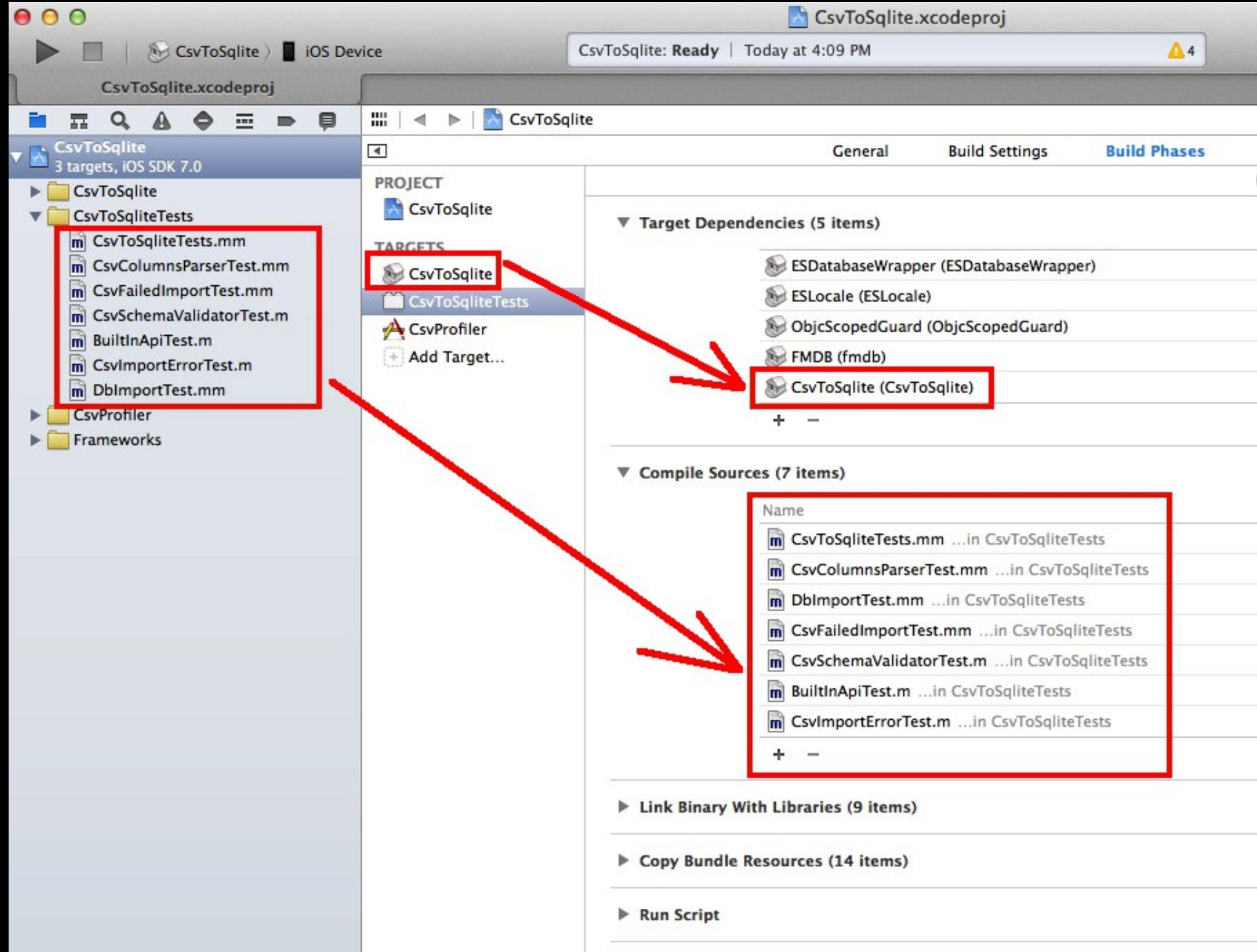


Never Link Libraries into Other Libraries





Same Approach for Unit Tests





All Testable Code Should be in libraries



Use Precompiled Headers

```
// Pure C headers go here
#include <math.h>

#ifndef __cplusplus
    // Pure C++ headers go here
    #include <vector>
#endif

#ifndef __OBJC__
    // Objective-C headers go here
    #import <Foundation/Foundation.h>

#ifndef __cplusplus
    #import "MyObjectiveCppClass.h"
#endif
#endif
```

But I do not Need C++

Print Message for Low Level Error

```
NSDictionary* errorMessages;  
NSLog( @"%@", errorMessages[ @( errorCode ) ] );
```

C++ ==> No Boxing

```
std::map< NSInteger, NSString* > errorMessages;  
NSLog( @"%@", errorMessages[ errorCode ] );
```

Scoped Guard

```
void bad(const char* p)
{
    FILE* fh = fopen(p,"r"); // acquire

    // use f
    if ( someCondition )
    {
        // Oops! File handle leaks
        return;
    }

    fclose( fh ); // release
}
```

```
void good(const char* p)
{
    FILE* fh = fopen(p,"r"); // acquire

    // the block to perform cleanup actions
    GuardCallbackBlock releaseBlock_ = ^void( void )
    {
        fclose( fh );
    };

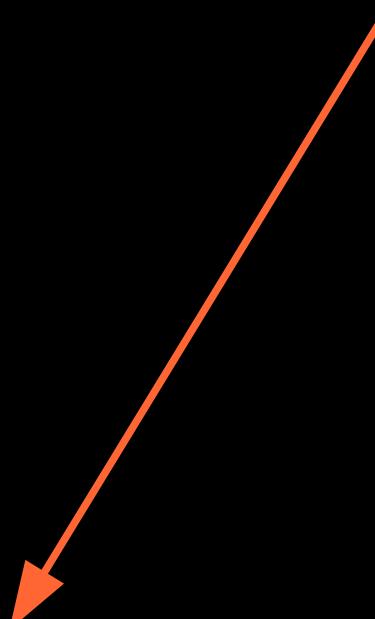
    // creating a guard
    ObjcScopedGuard guard( releaseBlock_ );

    if ( someCondition )
    {
        // Now the scoped guard will release the resource
        return;
    }
}
```

Exception-Safe Resource Deallocation

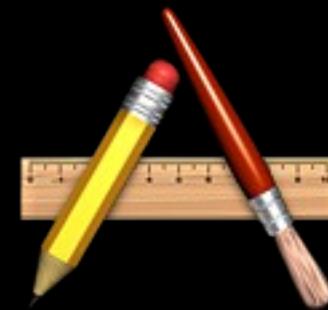
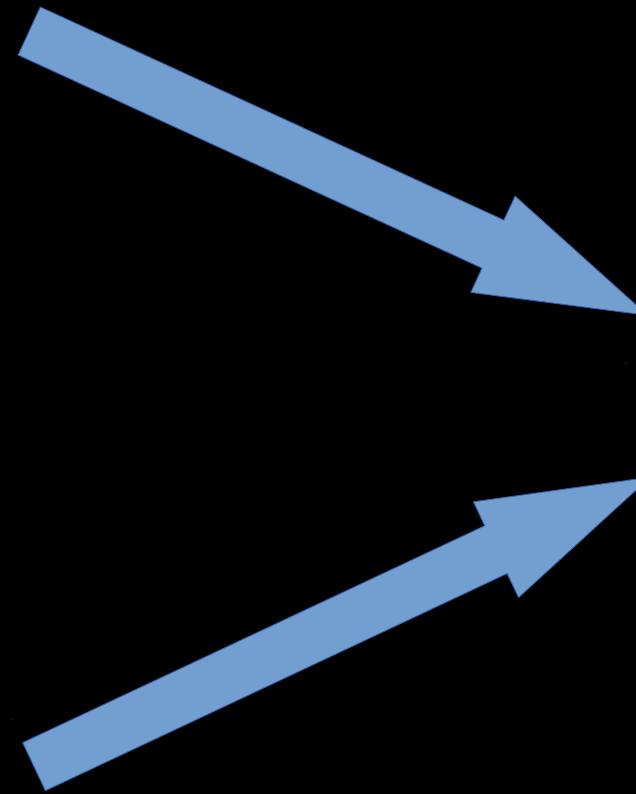
Vector as a Scoped Guard Memory

malloc()



std::vector<unsigned char>

Let's Recap





```
#import <AFNetworking/AFNetworking.h>
```

For Public Headers



For Development
and Unit Testing



For Distribution

Native Rulezzz

Alexander Dodatko

@dodikk88