iContinuousIntegration

by Oleksandr Dodatko
EPAM Systems

What's covered

Managing shared projects with xCode

Building a project without xCode GUI

Creating "universal binary" libraries
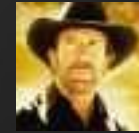
Deploying project and libraries for QA

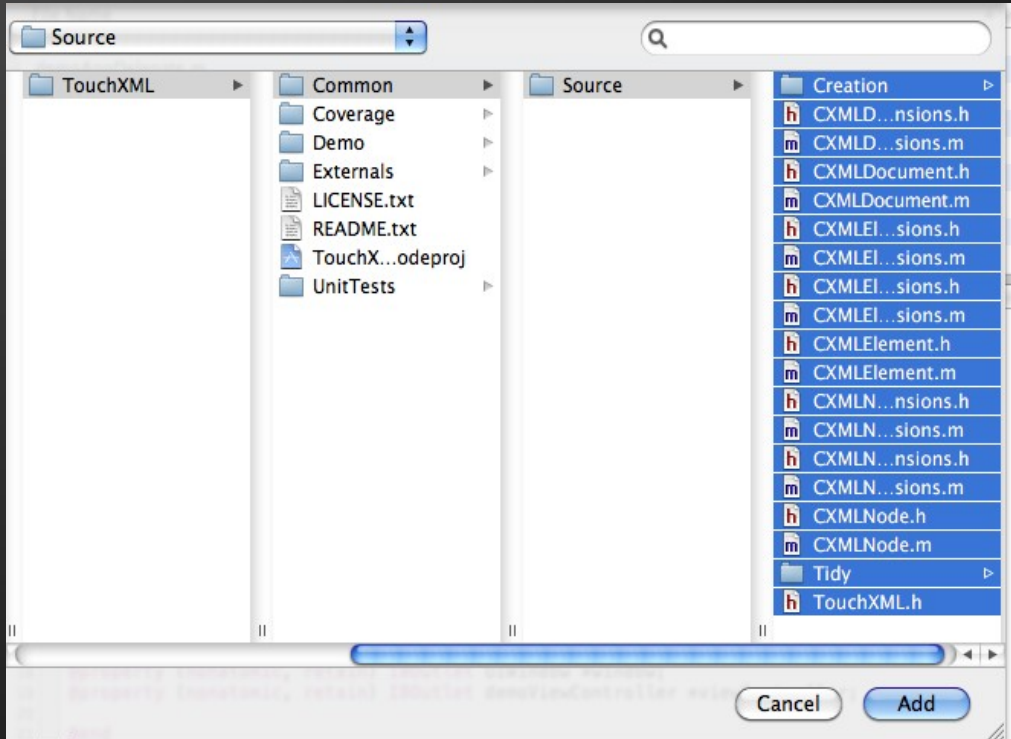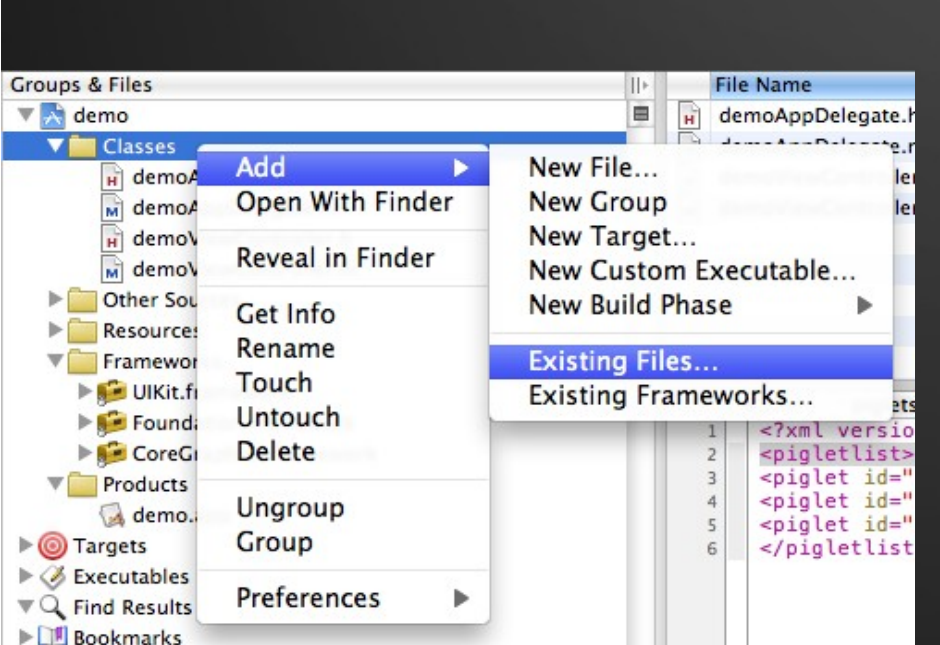# More FUN for developers

Unit testing with GHUnit

Using Hudson build server
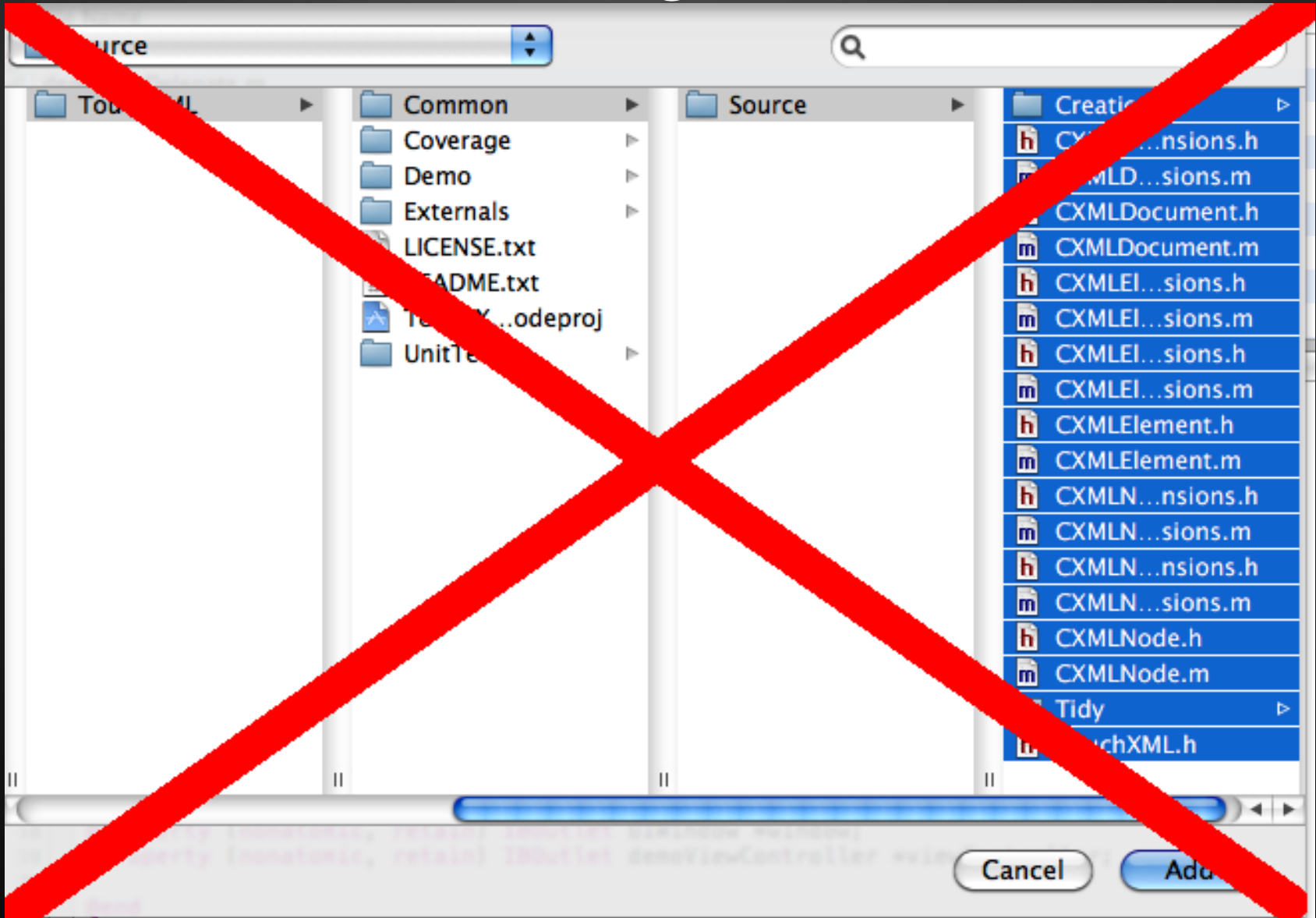(it has Chuck Norris plug-in)

Running applications on simulator
without xCode

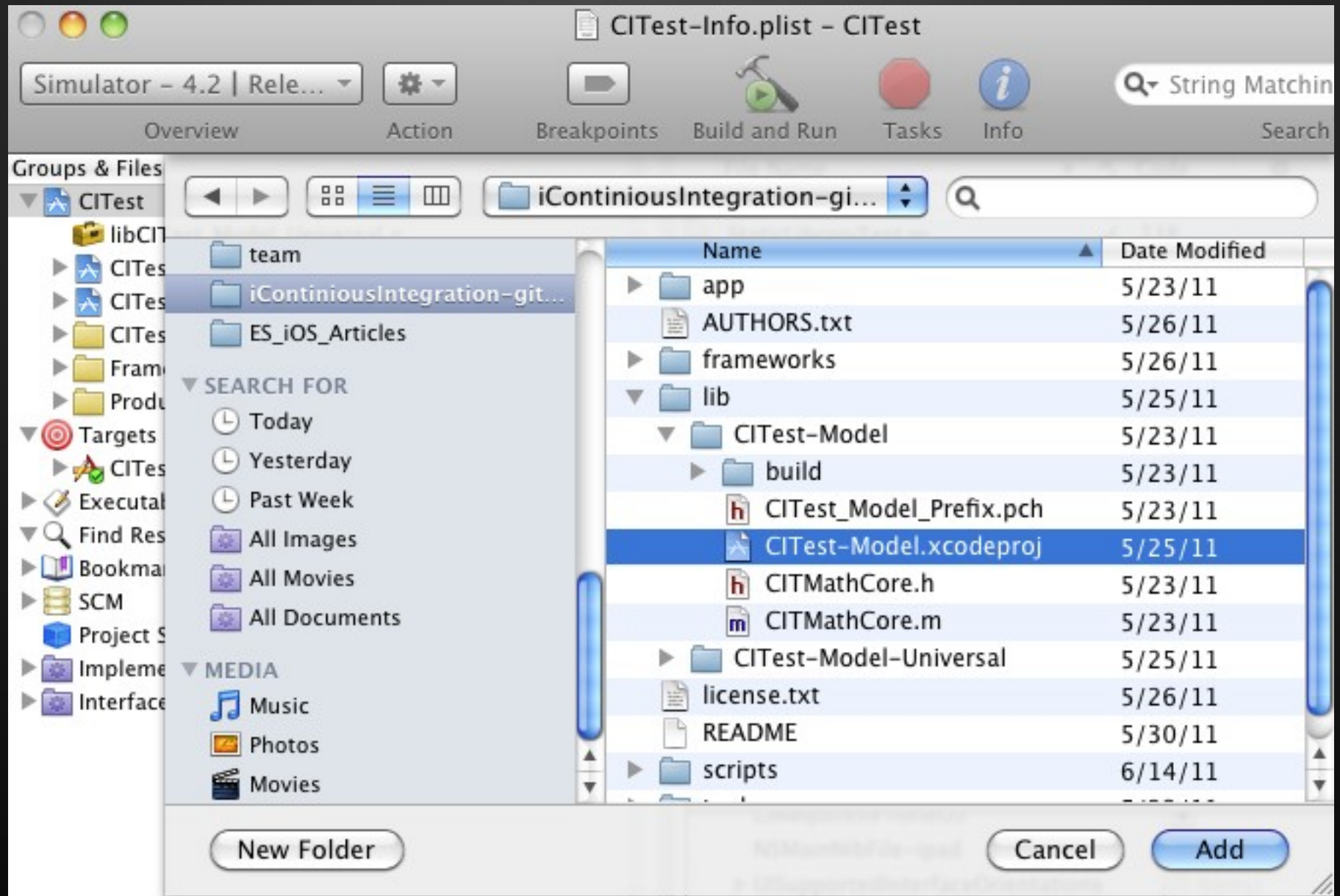# "Commonly used" Project organization

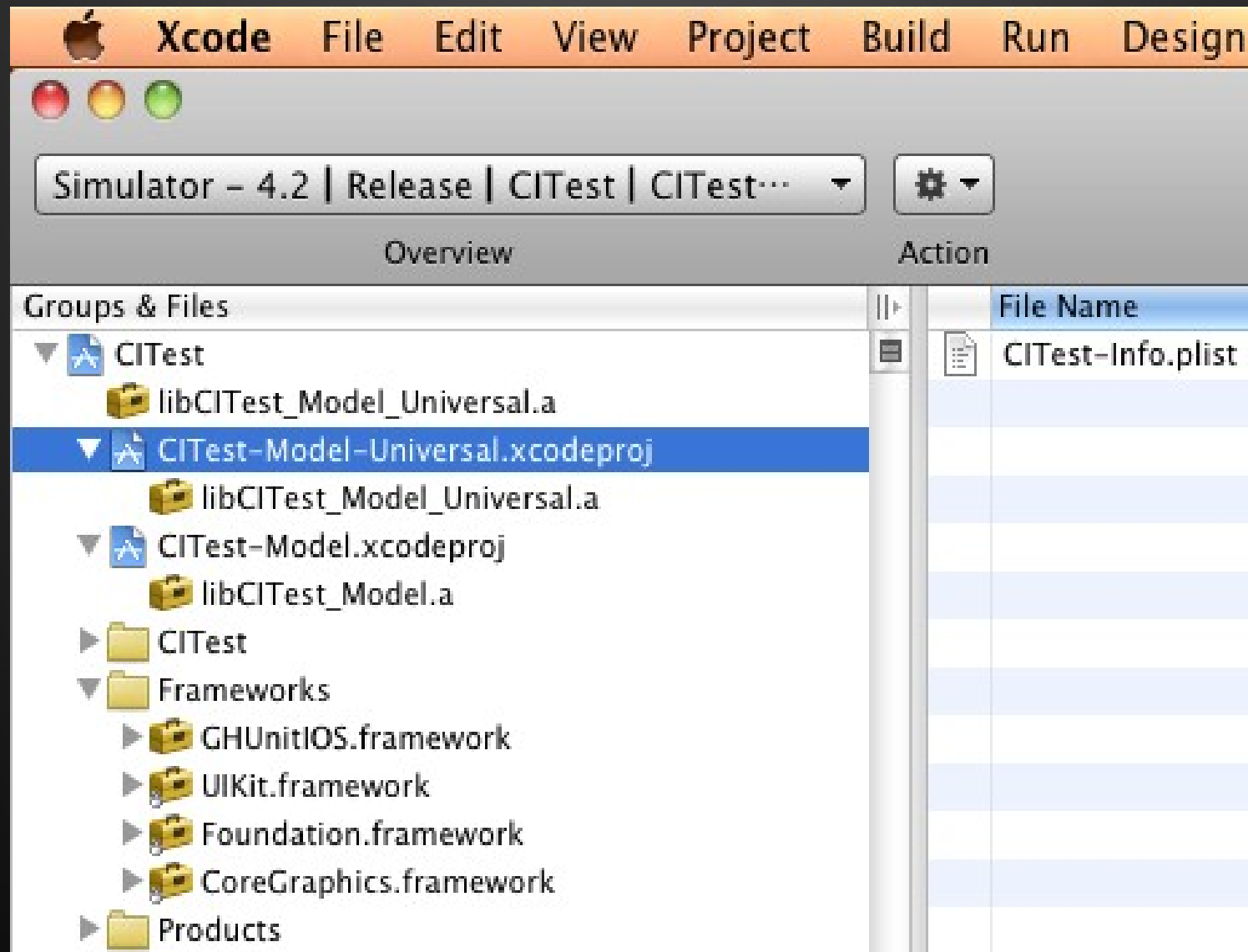# Wrong !!!

ONE Product,
ONE XCODE PROJECT

Library project howto

Library project howto

Setting up dependencies
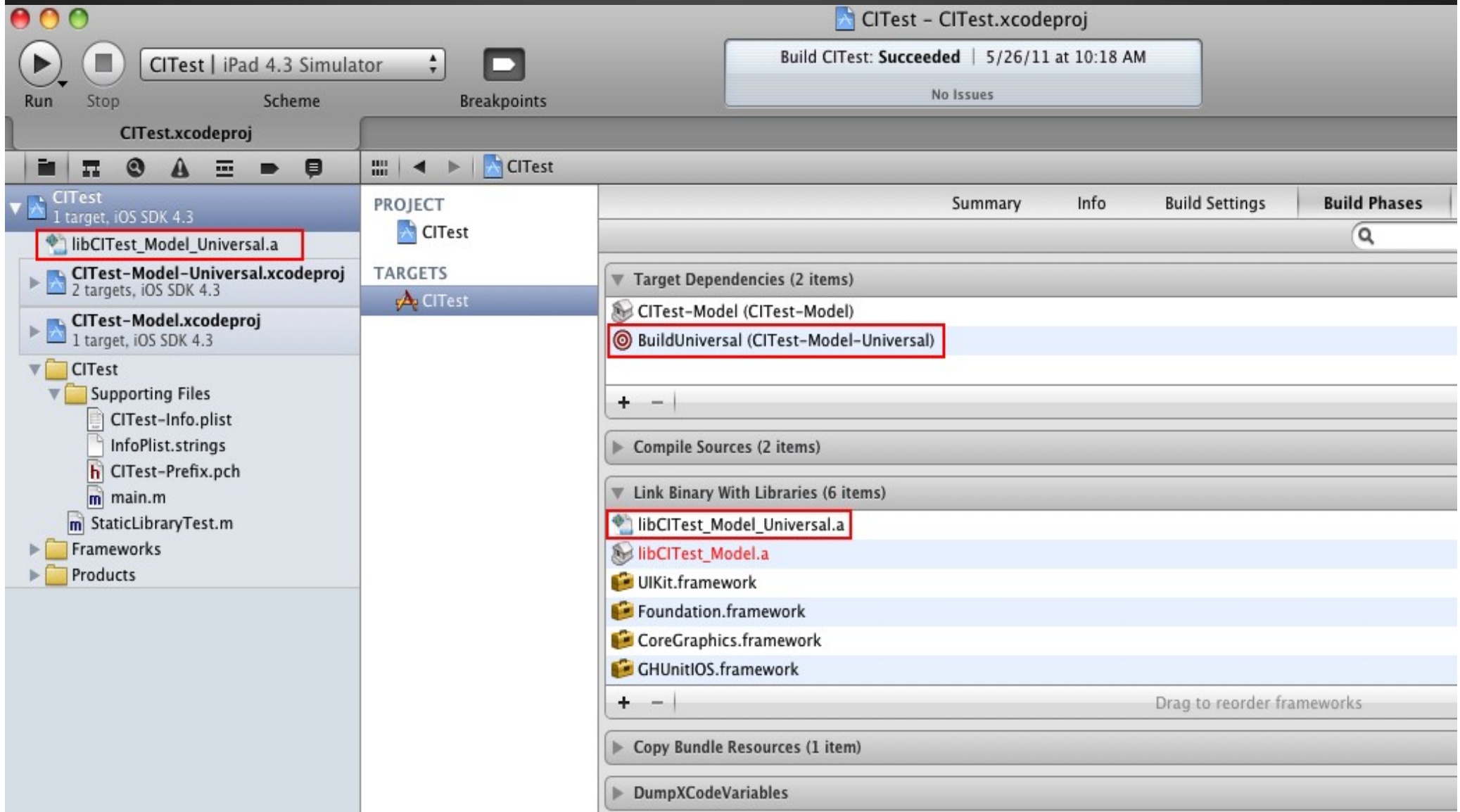
Defining project structure

app
lib
frameworks
scripts
tools
test
certificates
deployment
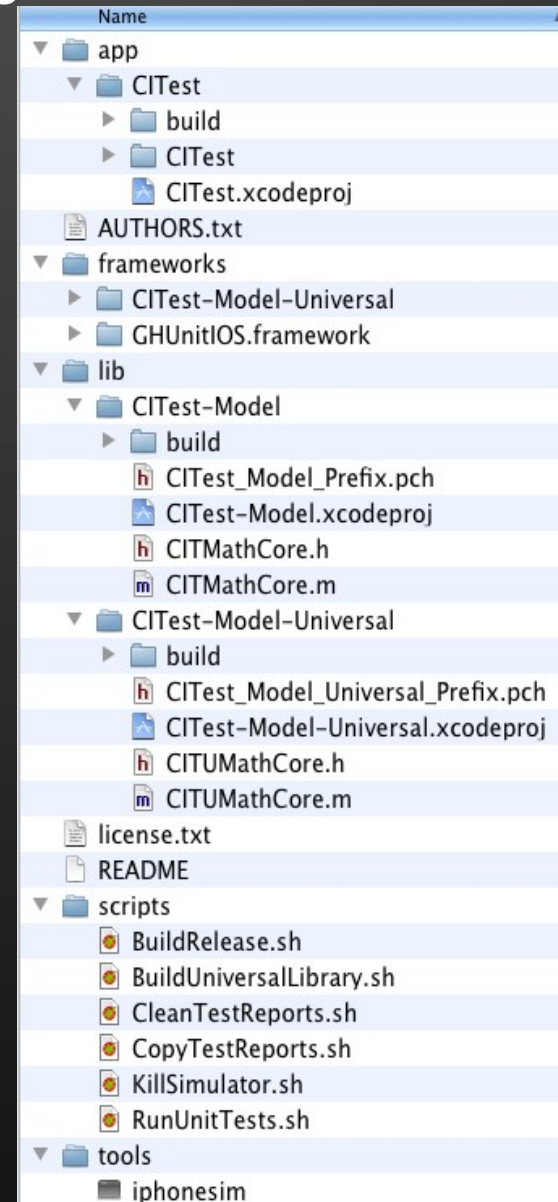
Deployment : Desktop vs. mobile

Desktop applications

Desktop applications

Mobile

QA

*.ipa

?

? ↑↓

Mobile

*.ipa
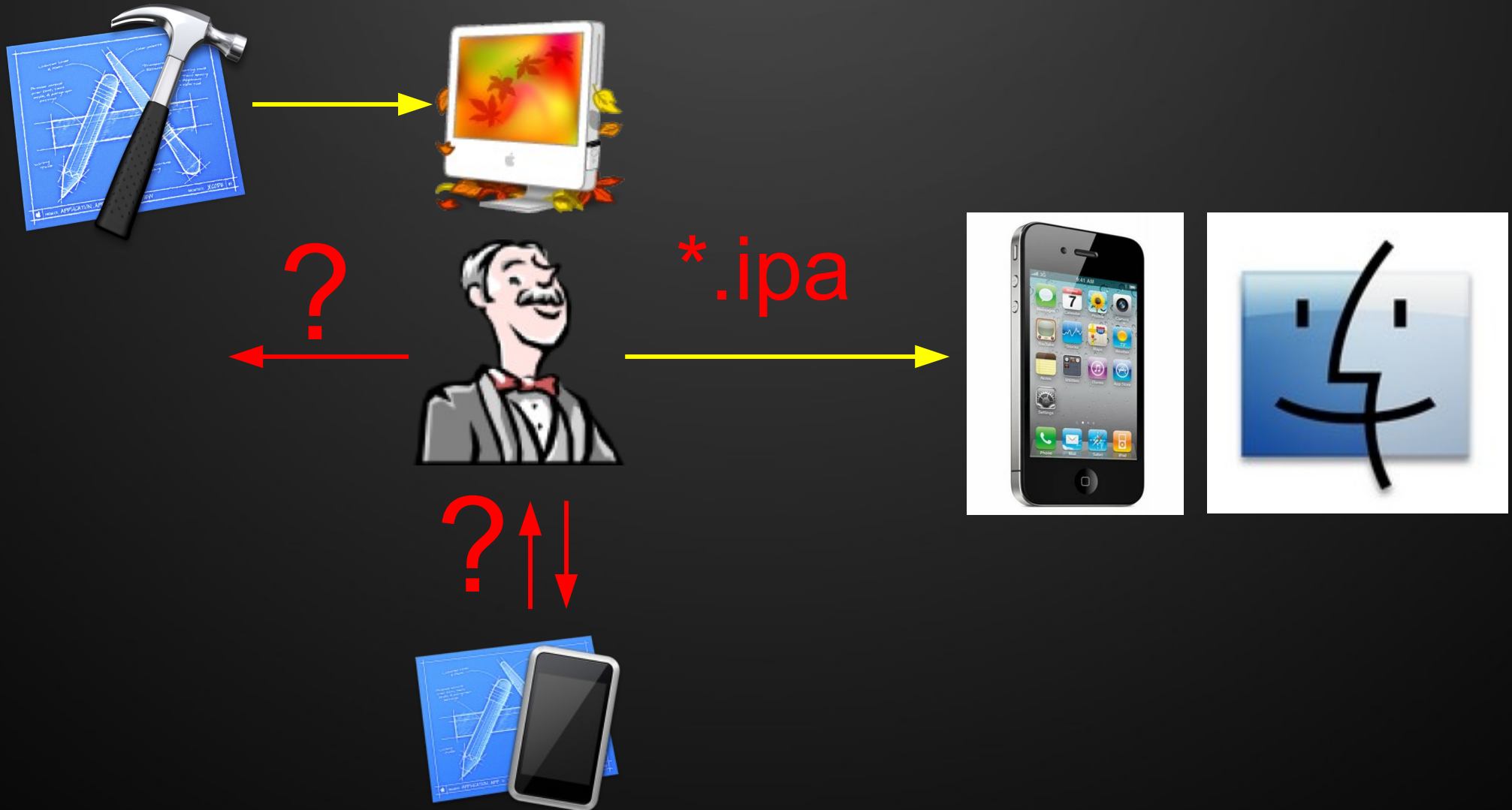
Mobile QA

•*.ipa

Building without xCode GUI

xcodebuild -project $PROJECT_NAME.xcodeproj

 -target $TARGET_NAME

-configuration Release

-parallelizeTargets

-sdk iphonesimulator4.3

clean build
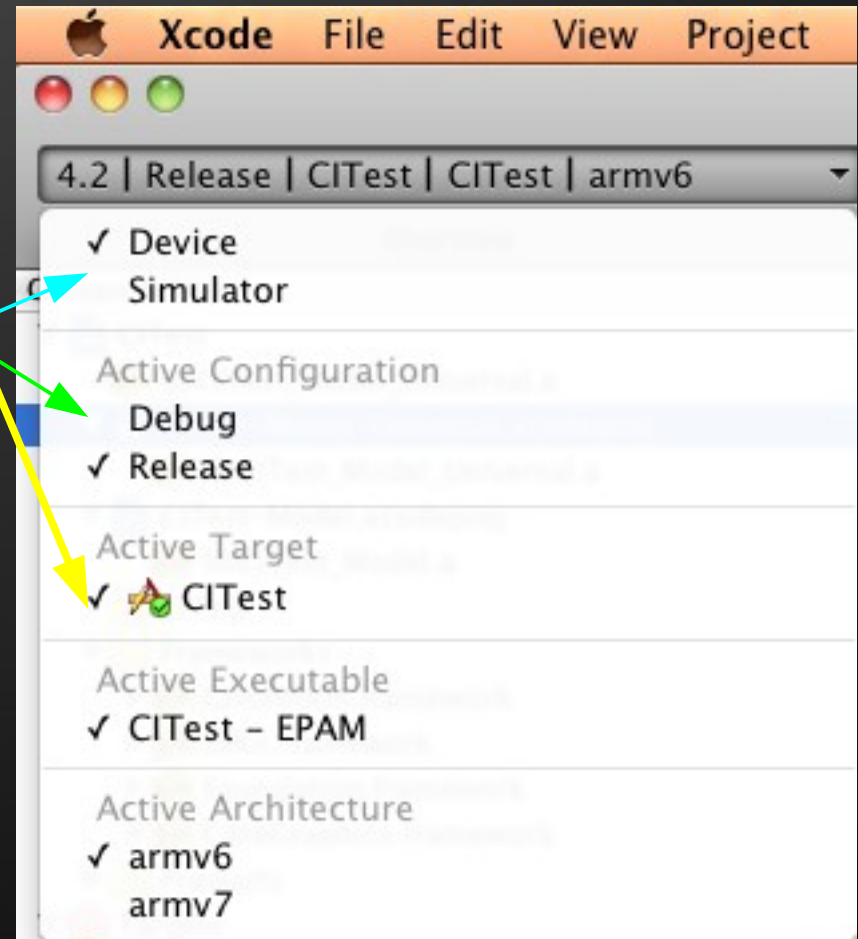
Creating *.ipa file

/usr/bin/xcrun

-sdk iphoneos PackageApplication
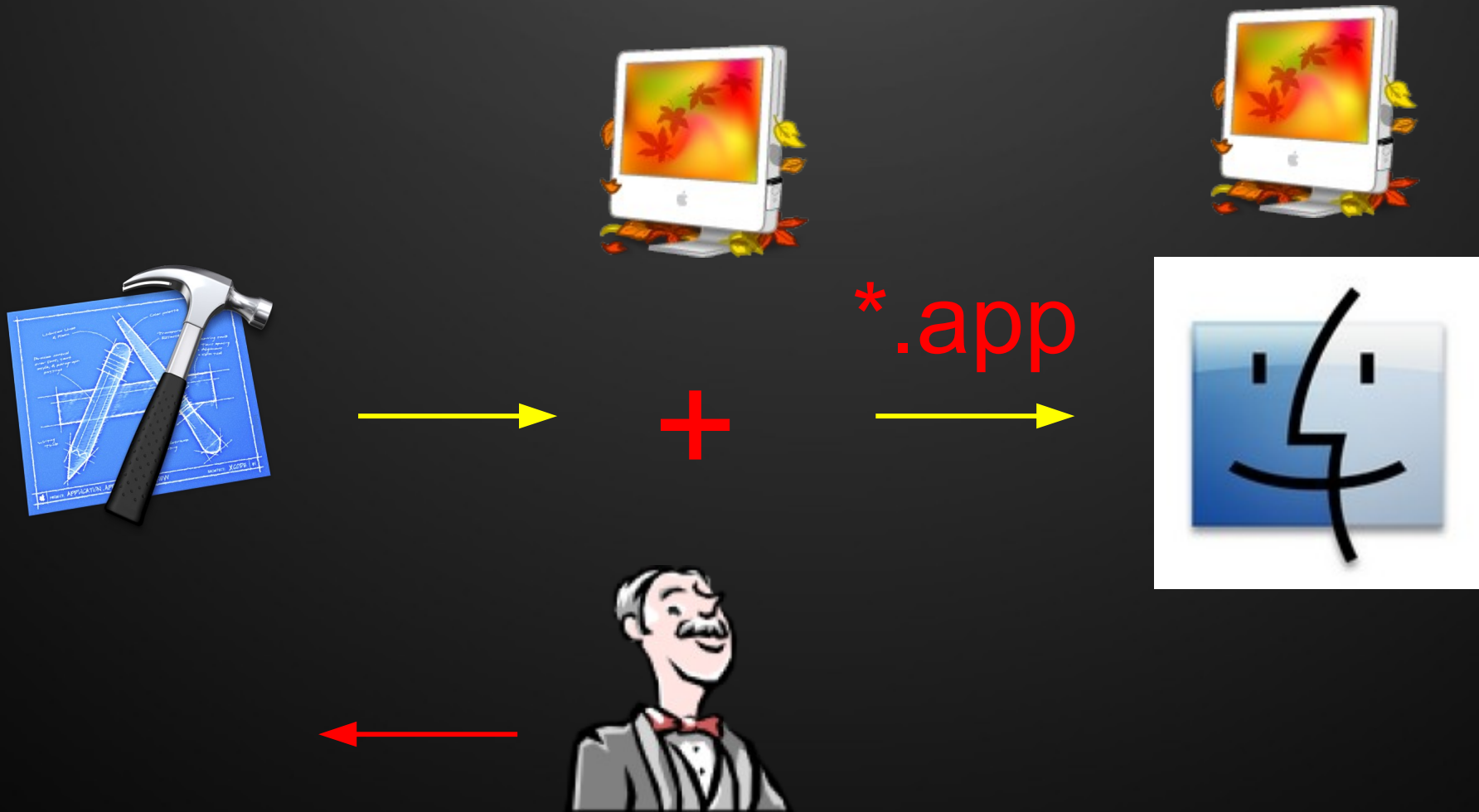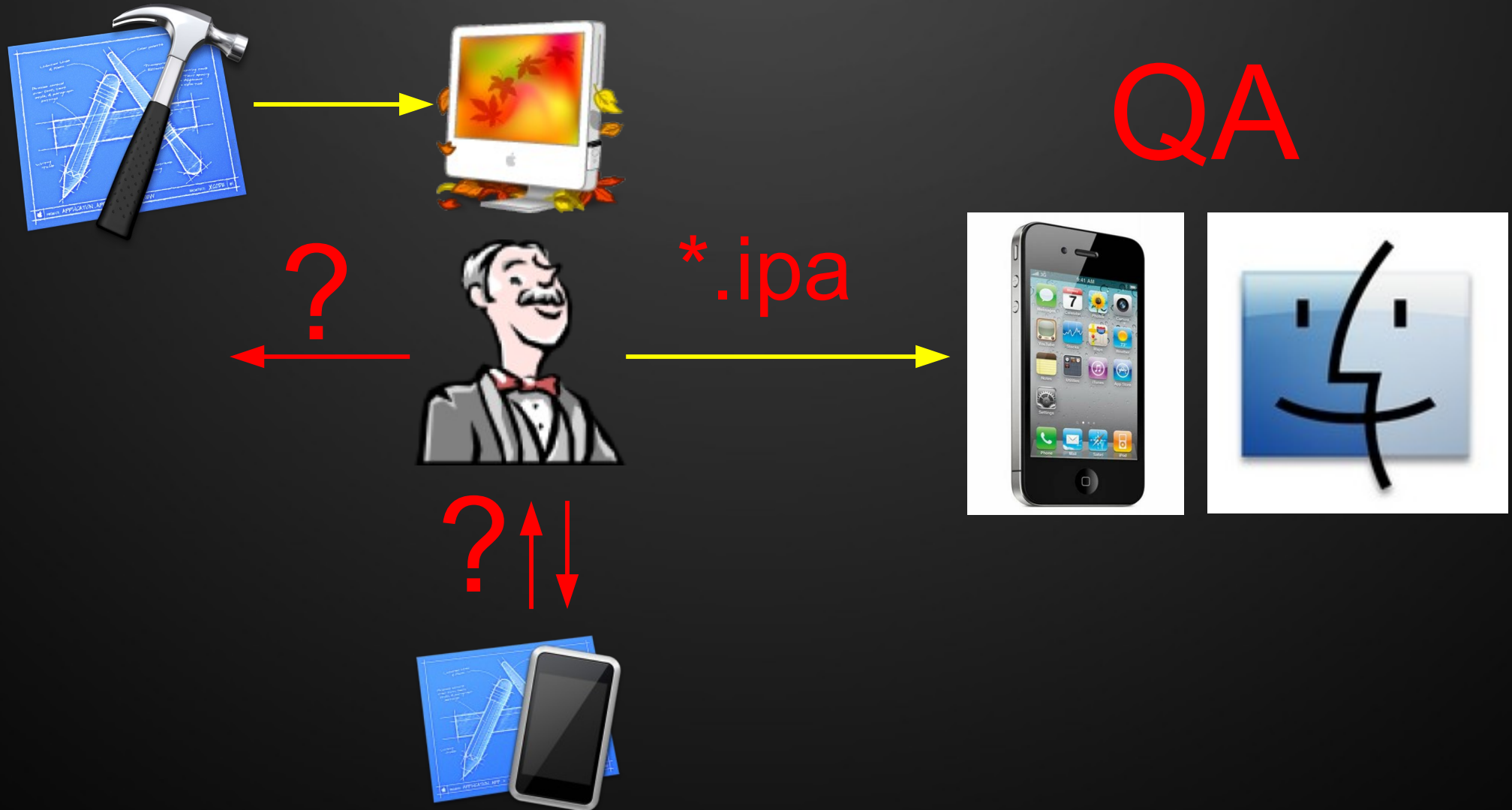
-v "${BUILD_DIR}/Release-iphoneos/$ {PROJECT_NAME}.app"

-o "${DEPLOYMENT_DIR}/$ {PROJECT_NAME}.ipa"

--sign "${DEVELOPER_NAME}"

--embed "${PROVISONING_PROFILE}"

Product	Window	Help
Run		⌘R
Test		⌘U
Profile		⌘I
Analyze		⇧⌘B
Archive		
Build For		▶
Perform Action		▶
Build		⌘B
Clean		⇧⌘K
Stop		⌘.
Debug		▶
Window Behavior		▶
Attach to Process		▶
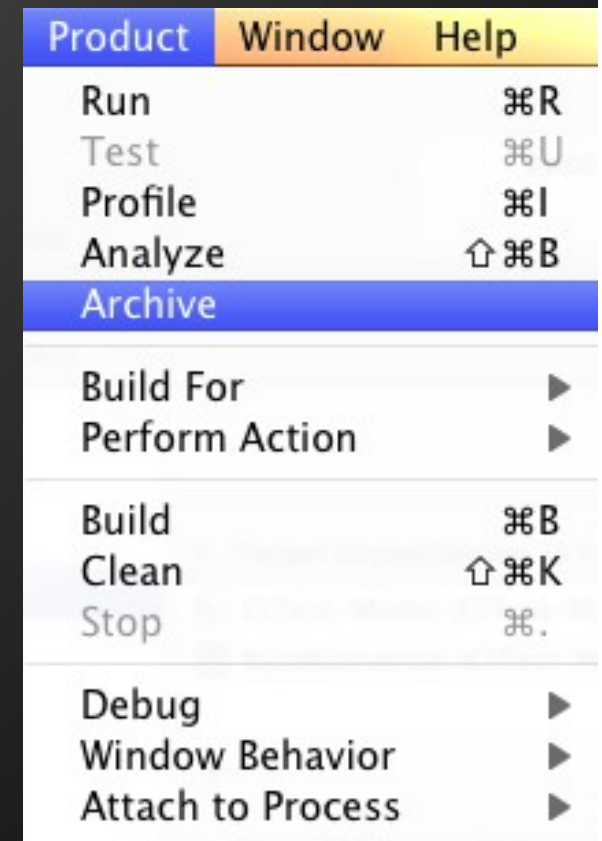
DEVELOPER_NAME="iPhone Developer: Oleksandr Dodatko (ABCDEFG123456)"

How about unit testing?

Picking a framework

Running a test

Collecting results

Test frameworks chart

	SenTest	**Google**	**GHUnit**
Xcode integration	+	+	---
UIKit Support	---	---	+
Bundles support	---	---	+
Xml reports	---	---	+ (lack of support for hudson CI)
Runs on device	+- (Runtime tests only)	+- (Runtime tests only)	+
Runs on simulator	+- (logic tests only)	+- (logic tests only)	+
Debugging (out of box)	---	---	+
UI snapshots comparing	---	+	---

GHUnit configuration

Add GHUnit.framework

Replace Main.h with the one from GHUnit

Remove "MainNibFile" entry from the info.plits

```
setenv( "GHUNIT_AUTORUN" , "YES", 1 );
setenv( "WRITE_JUNIT_XML", "YES", 1 );

// Not supported in the official GHUNIT
setenv( "GHUNIT_AUTOEXIT" , "YES", 1 );
```

Running test

"$TOOLS_DIR/iphonesim"

launch "$DEPLOYMENT_DIR/CITest.app"

4.2

ipad

NOTE : Use only FULL PATH to the app
as shown above

Collect results

```
TEMP_DIR=$(/usr/bin/getconf DARWIN_USER_TEMP_DIR)
TEST_DIR_NAME=test-results
TEST_RESULTS_DIR=$TEMP_DIR$TEST_DIR_NAME

## Now we can just copy test results
cd "$TEST_RESULTS_DIR"
  pwd
  cp *.xml "$TEST_PUBLISH_DIR"
cd "$LAUNCH_DIR"
```

Terminating simulator

killall -s -KILL -c "iphonesim"

killall -KILL -c "iphonesim"

killall -s -KILL -c "iPhone Simulator"

killall -KILL -c "iPhone Simulator"

Do it before you run a test app

Universal binaries

1. Build a library version for the device.

2. Build a library version for the simulator.

3. Combine them to a single binary

4. Deploy universal library to the "frameworks" directory.

Combining binaries

lipo -create

"${LIB_BUILD_DIR}/Release-iphoneos/libCITest_Model_Universal.a"

"${LIB_BUILD_DIR}/Release-iphonesimulator/libCITest_Model_Universal.a"

-output "../frameworks/CITest-Model-Universal/Lib/libCITest_Model_Universal.a"

Build script should

Build main products

Create *.ipa packages for main products

Build unit tests

Run unit tests with iphonesim

Package *.ipa and *.app entries to *.zip archive

Prepare unit test reports for deployment

Build server should

Checkout project sources

Run build script

Deploy product archives

Publish test reports

Contacts

EPAM systems (Dnipropetrovsk) http://www.epam.com/

Github page : **https://github.com/EmbeddedSources**
https://github.com/EmbeddedSources/iOS-articles

Oleksandr Dodatko – senior objC developer
mail/jabber   : dodikk88.reg@gmail.com
Skype   : alexander.dodatko.work@skype.com
Github page : **https://github.com/dodikk**