Continuous Improvement

Peter Robinett
2014-05-21

First, who am I?

I’m a mobile developer at Lua.

Also lots of freelance experience

A different CI

Continuous Improvement

The app is never done

What does this mean?

Your thinking evolves

The team evolves

The platform evolves

The app should evolve

How do you know your app needs improvement?

Code Smells

Crashes

Bad reviews

Duplicated code

Orphaned code

Old APIs

Mixed metaphors

Difficulties with tracing

Unnecessary dependencies

Where to start improving?

And more importantly, how to keep it up?

Documentation

Inline comments

VVDocumenter

appledoc

README.md

Monitoring

Downloads

Crashes

Usage

Testing

Unit tests

```
- (BOOL) isEven:
(NSNumber *)aNumber
```

```
@property BOOL isEven;

- (void) updateIsEven
```

UI tests

API tests

Why are they so damn hard?

Dealing with state

Mock it or skip it

Don’t test APIs, test how you use them

Test coverage

Test-writing discipline

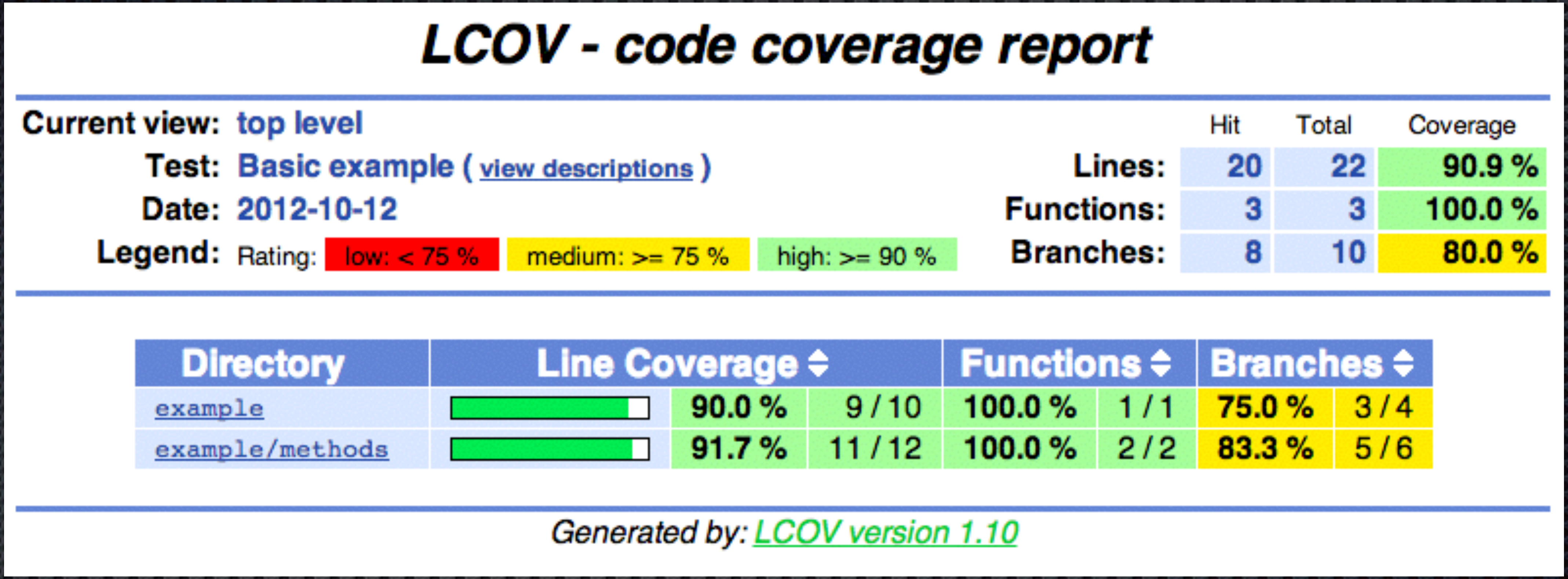
Write tests for bug fixes

Automation

Pick a service

- ~~Xcode CI~~
- Jenkins
- Travis

Publish reports



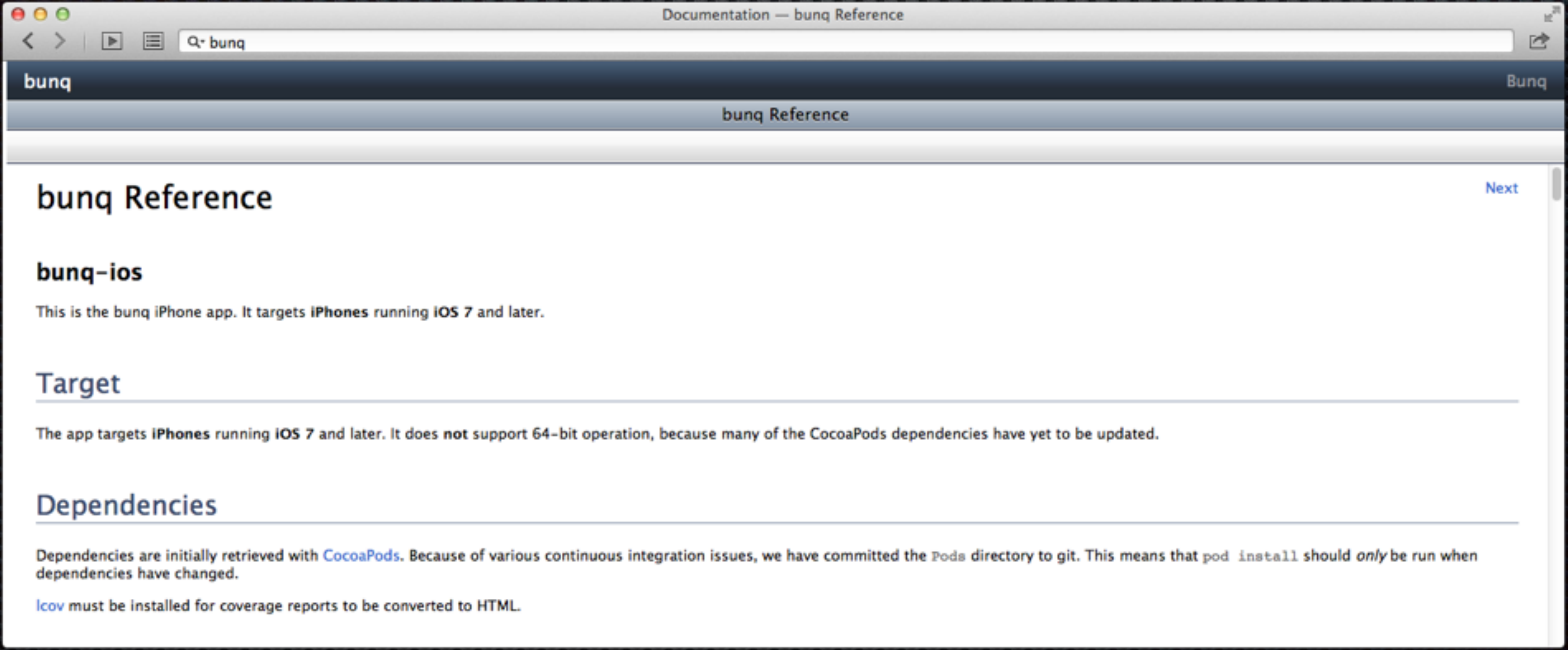
LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level				
Test:	Basic example (view descriptions)	Lines:	20	22	90.9 %
Date:	2012-10-12	Functions:	3	3	100.0 %
Legend:	Rating: low: < 75 % medium: >= 75 % high: >= 90 %	Branches:	8	10	80.0 %

Directory	Line Coverage ⬍			Functions ⬍		Branches ⬍	
example		90.0 %	9 / 10	100.0 %	1 / 1	75.0 %	3 / 4
example/methods		91.7 %	11 / 12	100.0 %	2 / 2	83.3 %	5 / 6

Generated by: LCOV version 1.10

```
# Upload coverage HTML pages to the server

IN_DIR="${OBJROOT}/myApp.build/Debug-iphonesimulator/myApp.build/Objects-
normal/i386"

# use the timestamp to create a unique dir
TS=`date +%s`
HUMAN_TS=`date`
OUT_DIR="/tmp/coverage/${TS}"
INTERMEDIATE_DIR="${OUT_DIR}_intermediate"
INTERMEDIATE_FILE_ALL="${OUT_DIR}_all.info"
INTERMEDIATE_FILE_FILTERED="${OUT_DIR}_filtered.info"

# copy all files so the permissions will be right for the current user
cp -R "${IN_DIR}" "${INTERMEDIATE_DIR}"

# get all values
/usr/local/bin/lcov -t "myApp coverage" -o "${INTERMEDIATE_FILE_ALL}" -c -d
"${INTERMEDIATE_DIR}" > /tmp/lcov_all.log 2>&1
# only extract ones for *myApp* files
/usr/local/bin/lcov -e "${INTERMEDIATE_FILE_ALL}" "*myApp*" -o "$
{INTERMEDIATE_FILE_FILTERED}" > /tmp/lcov_filtered.log 2>&1

# convert to HTML
/usr/local/bin/genhtml -q -s --legend -o "${OUT_DIR}" -t "Coverage for tests
run on ${HUMAN_TS}" "${INTERMEDIATE_FILE_FILTERED}" > /tmp/genhtml.log 2>&1

# put the files on the coverage server
scp -C -r "${OUT_DIR}" "coverage@coverage.myApp.com:/var/www/coverage/ios/$
{TS}"
```



bunq Reference

bunq-ios

This is the bunq iPhone app. It targets **iPhones** running **iOS 7** and later.

Target

The app targets **iPhones** running **iOS 7** and later. It does **not** support 64-bit operation, because many of the CocoaPods dependencies have yet to be updated.

Dependencies

Dependencies are initially retrieved with CocoaPods. Because of various continuous integration issues, we have committed the `Pods` directory to git. This means that `pod install` should *only* be run when dependencies have changed.

lcov must be installed for coverage reports to be converted to HTML.

```
# upload appledocs to the docs server
scp -C -r /tmp/appledoc_${USER}/${PROJECT_NAME}/html/*
"docs@docs.myApp.com:/var/www/docs/ios" > /tmp/docsupload.log 2>&1
```

Debugging

It is a skill

Try catching all exceptions

Instrumentation

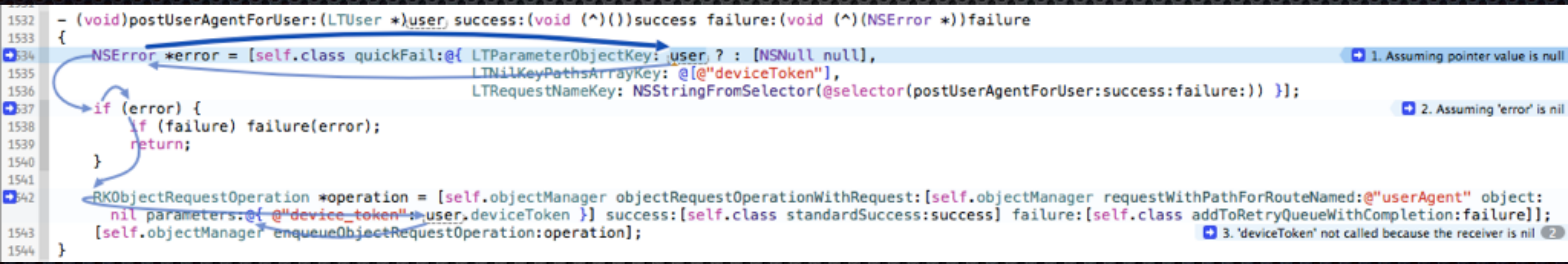
What to check?

- execution time
- memory usage
- memory (de)allocation, especially with Core Foundation
- threads/queues used
- network activity
- location services subscriptions

How to check?

Dumb timing with NSLog

Static analysis



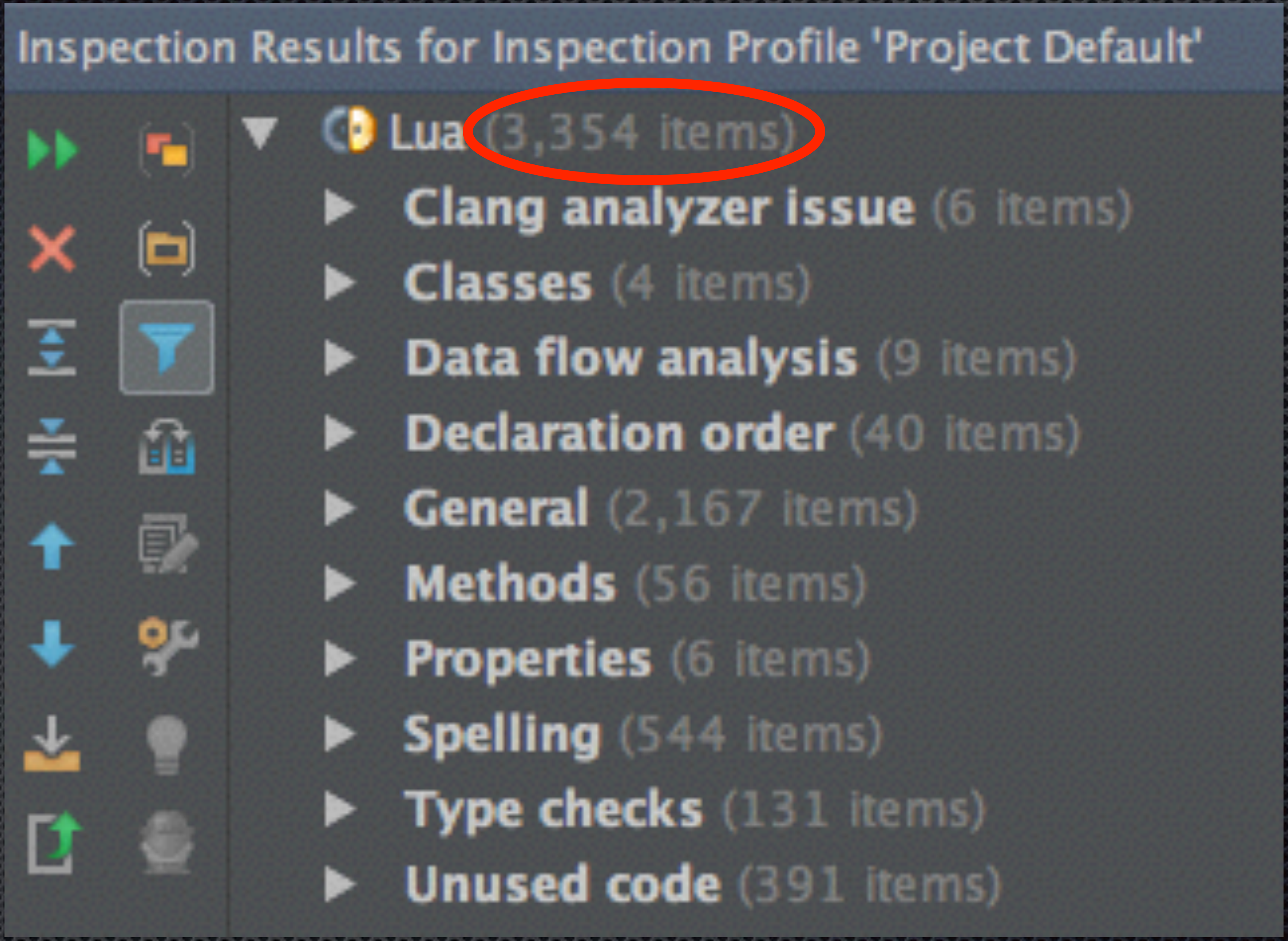
```
- (void)postUserAgentForUser:(LTUser *)user success:(void (^)())success failure:(void (^)(NSError *))failure
{
    NSError *error = [self.class quickFail:@{ LTParameterObjectKey: user ? : [NSNull null],
                                              LTNilKeyPathsArrayKey: @[@"deviceToken"],
                                              LTRequestNameKey: NSStringFromSelector(@selector(postUserAgentForUser:success:failure:)) }];
    if (error) {
        if (failure) failure(error);
        return;
    }

    RKObjectRequestOperation *operation = [self.objectManager objectRequestOperationWithRequest:[self.objectManager requestWithPathForRouteNamed:@"userAgent" object:
      nil parameters:@{ @"device_token": user.deviceToken }] success:[self.class standardSuccess:success] failure:[self.class addToRetryQueueWithCompletion:failure]];
    [self.objectManager enqueueObjectRequestOperation:operation];
}
```

1. Assuming pointer value is null

2. Assuming 'error' is nil

3. 'deviceToken' not called because the receiver is nil 2

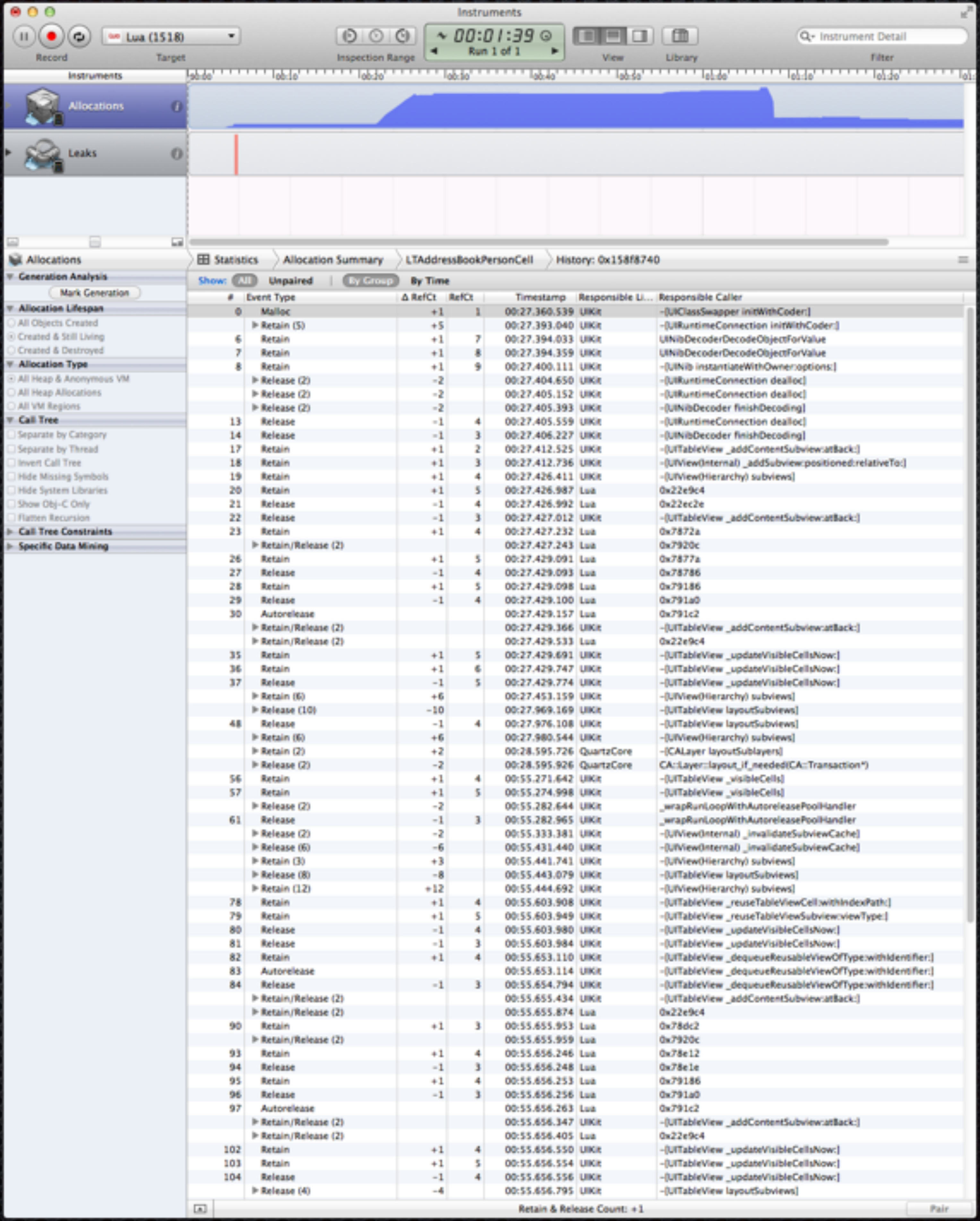
Static analysis with AppCode



Inspection Results for Inspection Profile 'Project Default'

```
▼ Lua (3,354 items)
    ▶ Clang analyzer issue (6 items)
    ▶ Classes (4 items)
    ▶ Data flow analysis (9 items)
    ▶ Declaration order (40 items)
    ▶ General (2,167 items)
    ▶ Methods (56 items)
    ▶ Properties (6 items)
    ▶ Spelling (544 items)
    ▶ Type checks (131 items)
    ▶ Unused code (391 items)
```

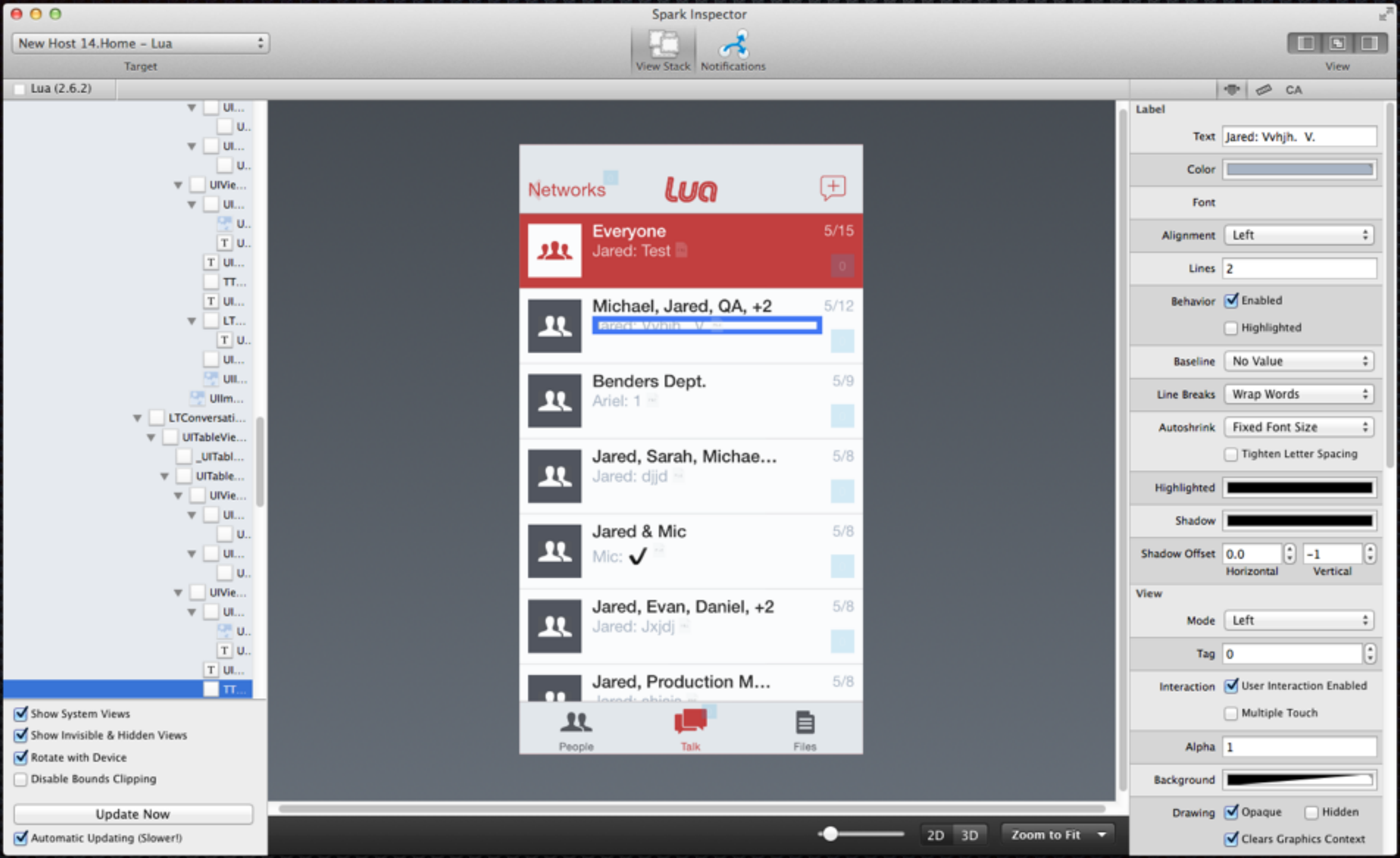
Instruments



#	Event Type	Δ RefCt	RefCt	Timestamp	Responsible Li...	Responsible Caller
0	Malloc	+1	1	00:27.360.539	UIKit	-[UIClassSwapper initWithCoder:]
	▶ Retain (5)	+5		00:27.393.040	UIKit	-[UIRuntimeConnection initWithCoder:]
6	Retain	+1	7	00:27.394.033	UIKit	UINibDecoderDecodeObjectForValue
7	Retain	+1	8	00:27.394.359	UIKit	UINibDecoderDecodeObjectForValue
8	Retain	+1	9	00:27.400.111	UIKit	-[UINib instantiateWithOwner:options:]
	▶ Release (2)	-2		00:27.404.650	UIKit	-[UIRuntimeConnection dealloc]
	▶ Release (2)	-2		00:27.405.152	UIKit	-[UIRuntimeConnection dealloc]
	▶ Release (2)	-2		00:27.405.393	UIKit	-[UINibDecoder finishDecoding]
13	Release	-1	4	00:27.405.559	UIKit	-[UIRuntimeConnection dealloc]
14	Release	-1	3	00:27.406.227	UIKit	-[UINibDecoder finishDecoding]
17	Retain	+1	2	00:27.412.525	UIKit	-[UITableView _addContentSubview:atBack:]
18	Retain	+1	3	00:27.412.736	UIKit	-[UIView(Internal) _addSubview:positioned:relativeTo:]
19	Retain	+1	4	00:27.426.411	UIKit	-[UIView(Hierarchy) subviews]
20	Retain	+1	5	00:27.426.987	Lua	0x22e9c4
21	Release	-1	4	00:27.426.992	Lua	0x22ec2e
22	Release	-1	3	00:27.427.012	UIKit	-[UITableView _addContentSubview:atBack:]
23	Retain	+1	4	00:27.427.232	Lua	0x7872a
	▶ Retain/Release (2)			00:27.427.243	Lua	0x7920c
26	Retain	+1	5	00:27.429.091	Lua	0x7877a
27	Release	-1	4	00:27.429.093	Lua	0x78786
28	Retain	+1	5	00:27.429.098	Lua	0x79186
29	Release	-1	4	00:27.429.100	Lua	0x791a0
30	Autorelease			00:27.429.157	Lua	0x791c2
	▶ Retain/Release (2)			00:27.429.366	UIKit	-[UITableView _addContentSubview:atBack:]
	▶ Retain/Release (2)			00:27.429.533	Lua	0x22e9c4
35	Retain	+1	5	00:27.429.691	UIKit	-[UITableView _updateVisibleCellsNow:]
36	Retain	+1	6	00:27.429.747	UIKit	-[UITableView _updateVisibleCellsNow:]
37	Release	-1	5	00:27.429.774	UIKit	-[UITableView _updateVisibleCellsNow:]
	▶ Retain (6)	+6		00:27.453.159	UIKit	-[UIView(Hierarchy) subviews]
	▶ Release (10)	-10		00:27.969.169	UIKit	-[UITableView layoutSubviews]
48	Release	-1	4	00:27.976.108	UIKit	-[UITableView layoutSubviews]
	▶ Retain (6)	+6		00:27.980.544	UIKit	-[UIView(Hierarchy) subviews]
	▶ Retain (2)	+2		00:28.595.726	QuartzCore	-[CALayer layoutSublayers]
	▶ Release (2)	-2		00:28.595.926	QuartzCore	CA::Layer::layout_if_needed(CA::Transaction*)
56	Retain	+1	4	00:55.271.642	UIKit	-[UITableView _visibleCells]
57	Retain	+1	5	00:55.274.998	UIKit	-[UITableView _visibleCells]
	▶ Release (2)	-2		00:55.282.644	UIKit	_wrapRunLoopWithAutoreleasePoolHandler
61	Release	-1	3	00:55.282.965	UIKit	_wrapRunLoopWithAutoreleasePoolHandler
	▶ Release (2)	-2		00:55.333.381	UIKit	-[UIView(Internal) _invalidateSubviewCache]
	▶ Release (6)	-6		00:55.431.440	UIKit	-[UIView(Internal) _invalidateSubviewCache]
	▶ Retain (3)	+3		00:55.441.741	UIKit	-[UIView(Hierarchy) subviews]
	▶ Release (8)	-8		00:55.443.079	UIKit	-[UITableView layoutSubviews]
	▶ Retain (12)	+12		00:55.444.692	UIKit	-[UIView(Hierarchy) subviews]
78	Retain	+1	4	00:55.603.908	UIKit	-[UITableView _reuseTableViewCell:withIndexPath:]
79	Retain	+1	5	00:55.603.949	UIKit	-[UITableView _reuseTableViewSubview:viewType:]
80	Release	-1	4	00:55.603.980	UIKit	-[UITableView _updateVisibleCellsNow:]
81	Release	-1	3	00:55.603.984	UIKit	-[UITableView _updateVisibleCellsNow:]
82	Retain	+1	4	00:55.653.110	UIKit	-[UITableView _dequeueReusableViewOfType:withIdentifier:]
83	Autorelease			00:55.653.114	UIKit	-[UITableView _dequeueReusableViewOfType:withIdentifier:]
84	Release	-1	3	00:55.654.794	UIKit	-[UITableView _dequeueReusableViewOfType:withIdentifier:]
	▶ Retain/Release (2)			00:55.655.434	UIKit	-[UITableView _addContentSubview:atBack:]
	▶ Retain/Release (2)			00:55.655.874	Lua	0x22e9c4
90	Retain	+1	3	00:55.655.953	Lua	0x78dc2
	▶ Retain/Release (2)			00:55.655.959	Lua	0x7920c
93	Retain	+1	4	00:55.656.246	Lua	0x78e12
94	Release	-1	3	00:55.656.248	Lua	0x78e1e
95	Retain	+1	4	00:55.656.253	Lua	0x79186
96	Release	-1	3	00:55.656.256	Lua	0x791a0
97	Autorelease			00:55.656.263	Lua	0x791c2
	▶ Retain/Release (2)			00:55.656.347	UIKit	-[UITableView _addContentSubview:atBack:]
	▶ Retain/Release (2)			00:55.656.405	Lua	0x22e9c4
102	Retain	+1	4	00:55.656.550	UIKit	-[UITableView _updateVisibleCellsNow:]
103	Retain	+1	5	00:55.656.554	UIKit	-[UITableView _updateVisibleCellsNow:]
104	Release	-1	4	00:55.656.556	UIKit	-[UITableView _updateVisibleCellsNow:]
	▶ Release (4)	-4		00:55.656.795	UIKit	-[UITableView layoutSubviews]

Retain & Release Count: +1

Spark Inspector



[✓] Show System Views
[✓] Show Invisible & Hidden Views
[✓] Rotate with Device
[ ] Disable Bounds Clipping
[✓] Automatic Updating (Slower!)

Label

Text	Jared: Vvhjh. V.
Color	
Font	
Alignment	Left
Lines	2
Behavior	[✓] Enabled [ ] Highlighted
Baseline	No Value
Line Breaks	Wrap Words
Autoshrink	Fixed Font Size [ ] Tighten Letter Spacing
Highlighted	■
Shadow	■
Shadow Offset	0.0 Horizontal, -1 Vertical

View

Mode	Left
Tag	0
Interaction	[✓] User Interaction Enabled [ ] Multiple Touch
Alpha	1
Background	
Drawing	[✓] Opaque [ ] Hidden [✓] Clears Graphics Context

Distribution

- TestFlight
- Hockey App
- etc

iOS Enterprise Account

Simulator apps

Thank you

Any questions?

I’m happy to answer any questions later too.

You can email me at peter@getlua.com.

I’m @pr1001 on Twitter.