

# Compiling & Optimizing Your Own Browser with WebKit

**ARIYA HIDAYAT**  
ENGINEERING DIRECTOR, SENCHA



# whoami



# Sencha

TROLLTECH®



NOKIA

QUALCOMM®

# Overview



# WebKit Everywhere

Browser



Devices



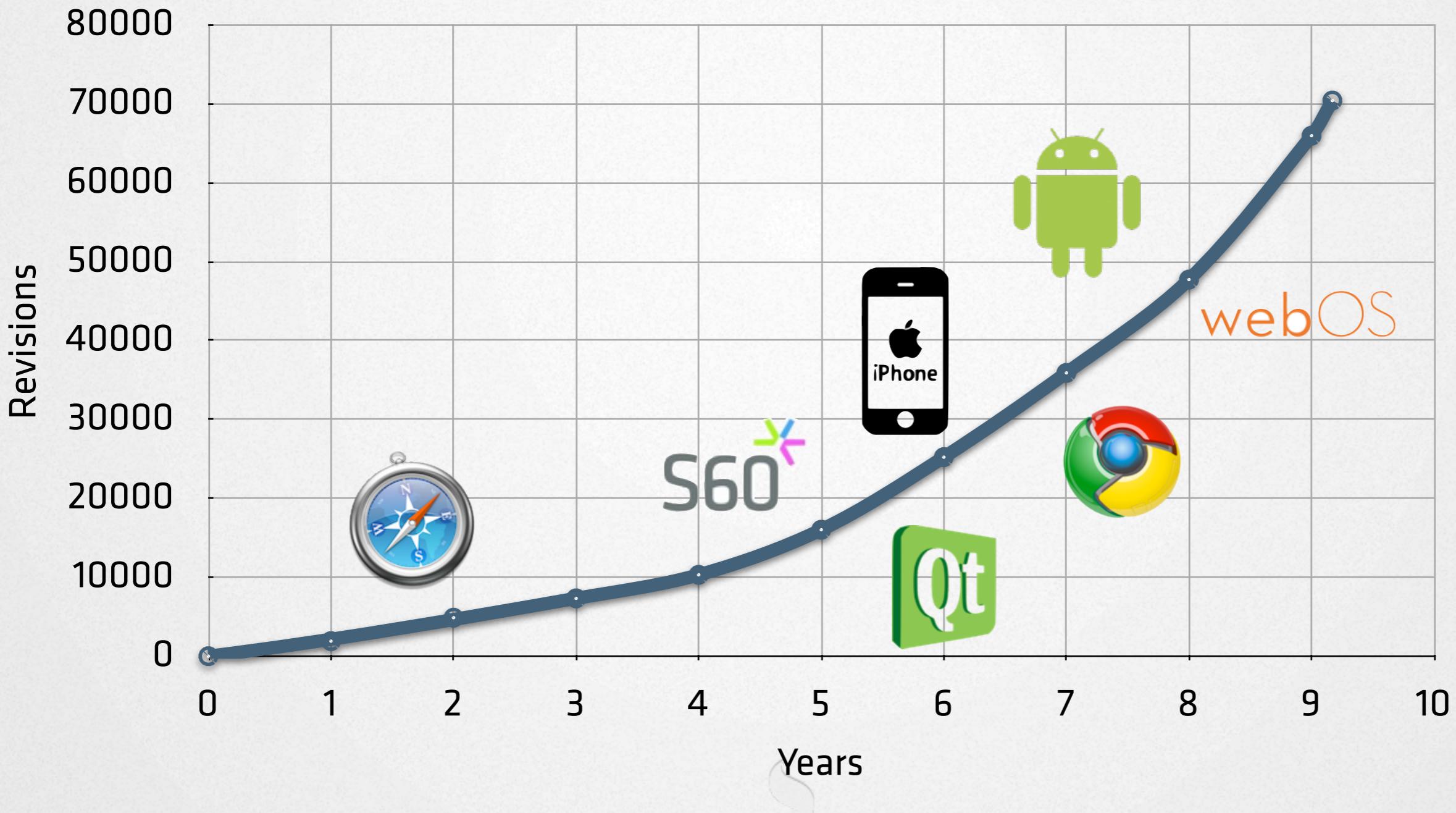
symbian

webOS

Runtime



# History



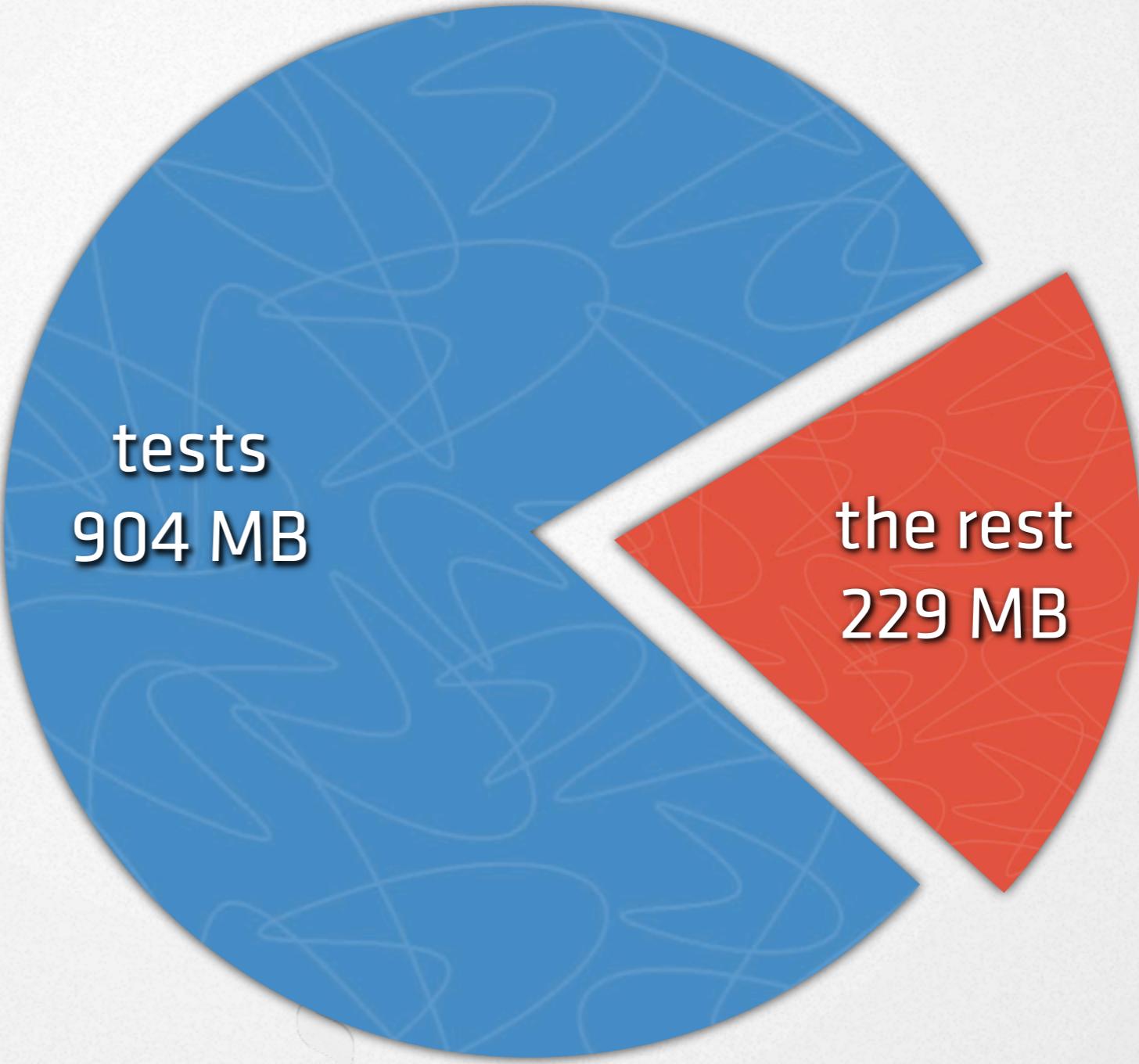
# History

~2000 commits/month



# Extensive Tests

≈ 20,000 tests



tests  
904 MB

the rest  
229 MB

# Workflow

1

Every commit needs to be reviewed

2

Broken commit must be reverted



# Workflow



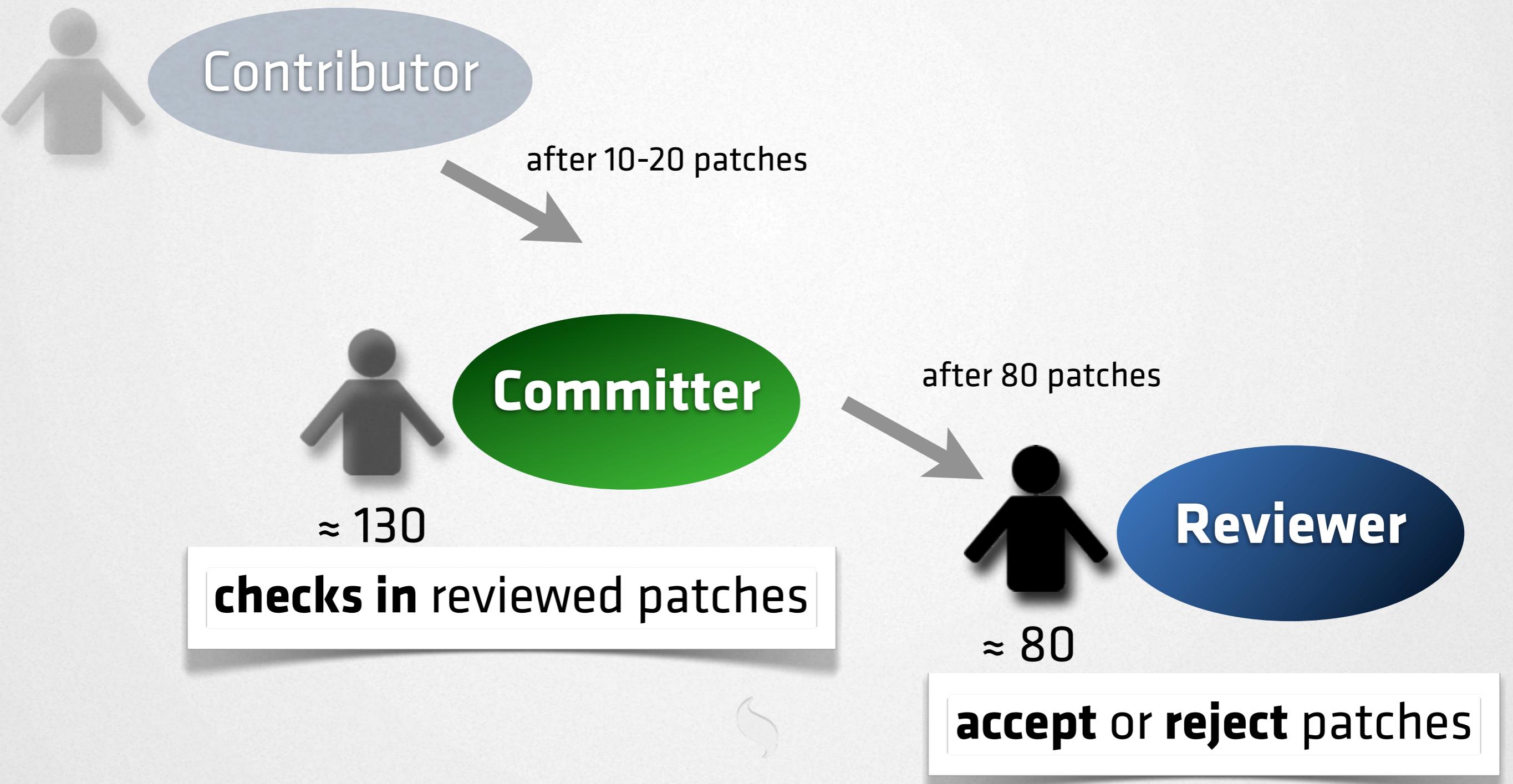
quality control



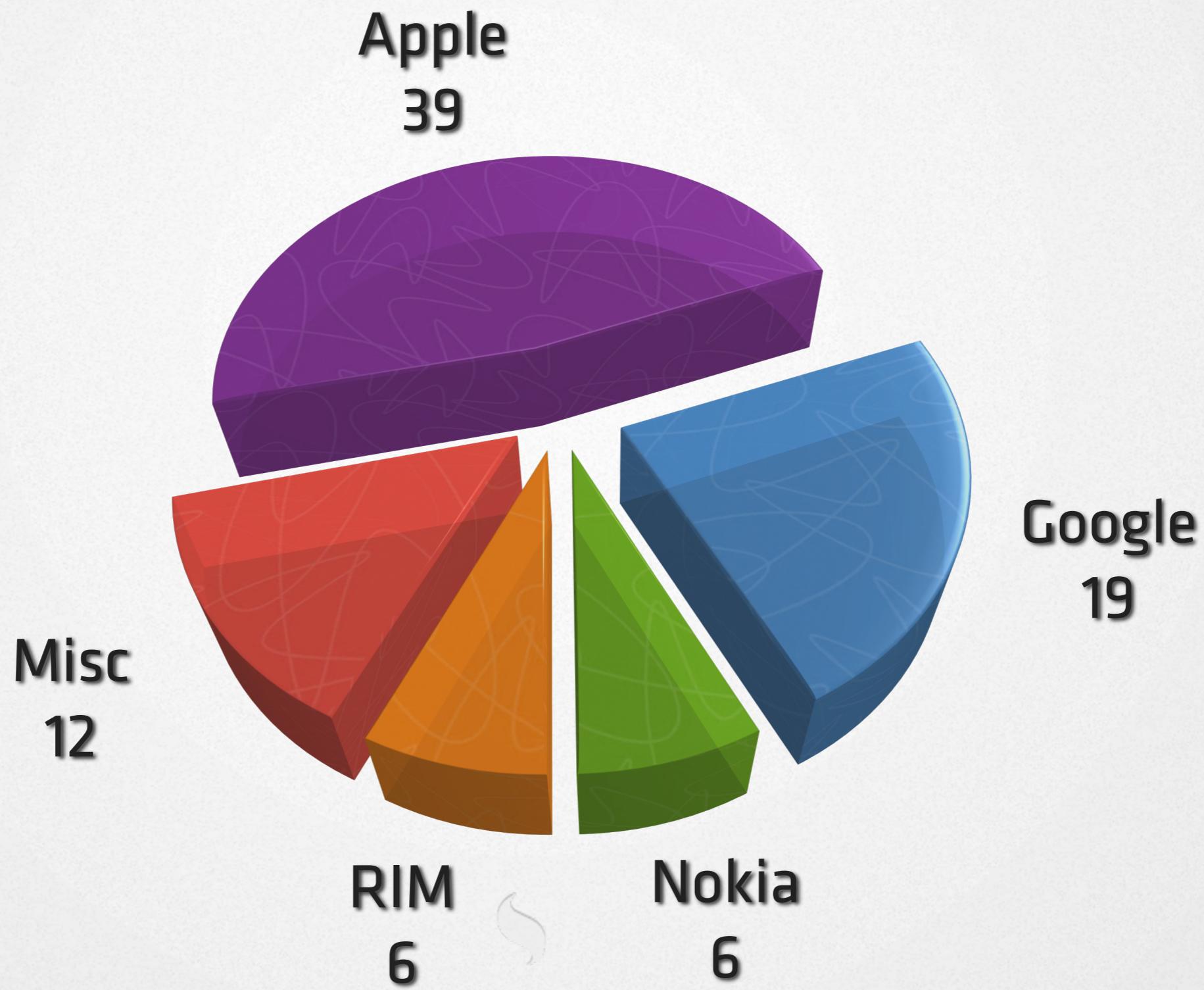
zero-regression policy



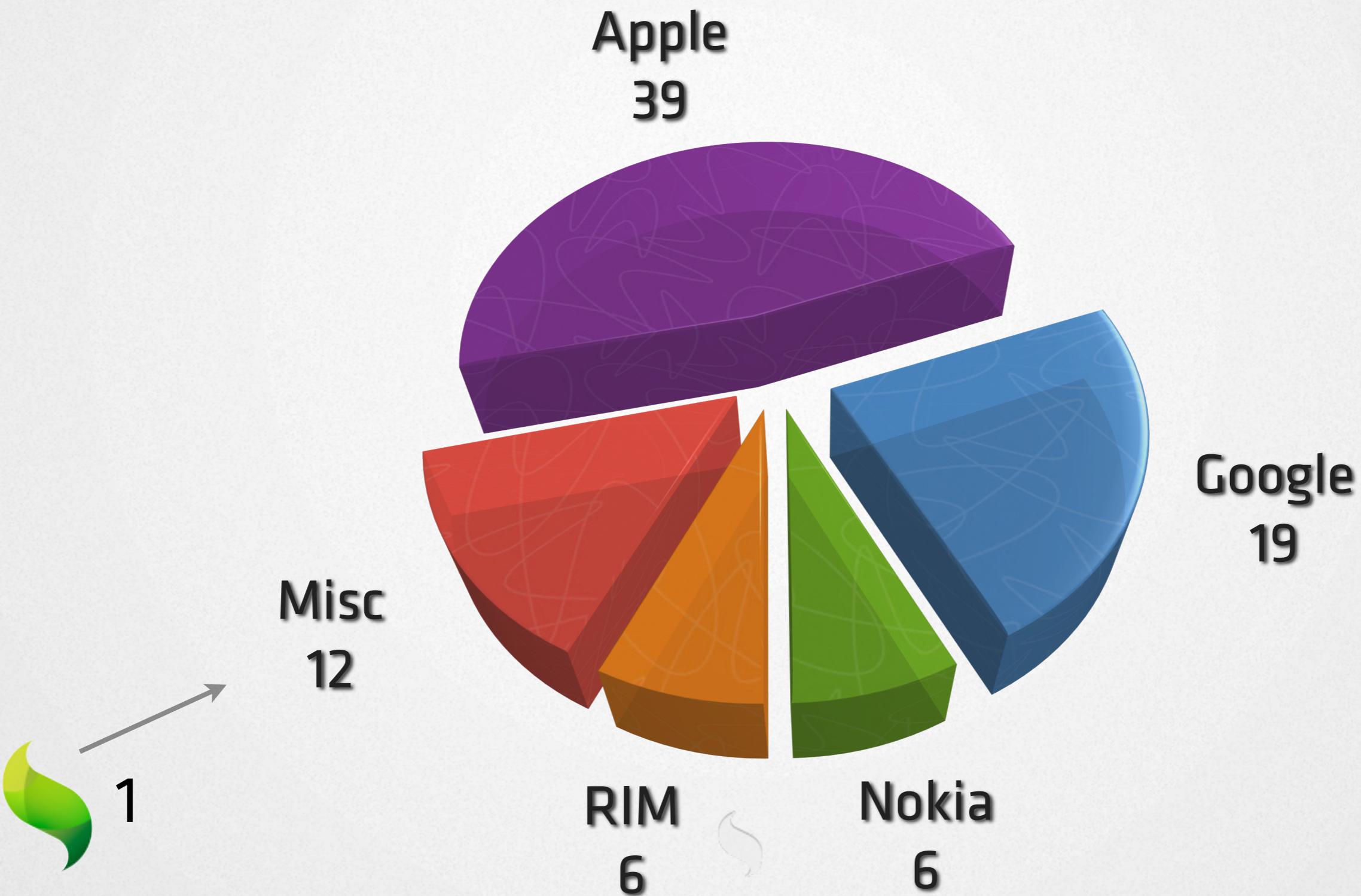
# Level of Involvement



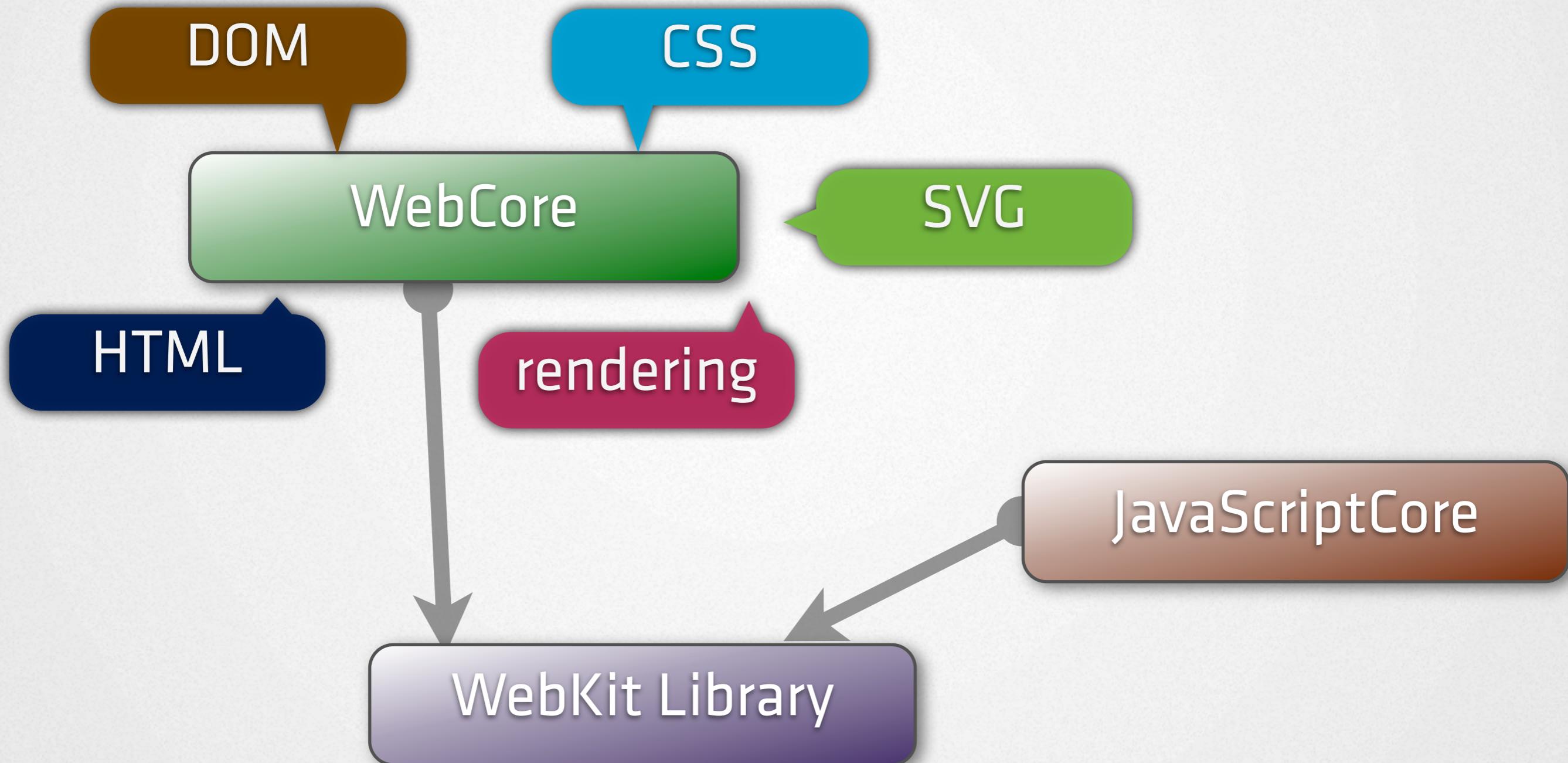
# WebKit Reviewers



# WebKit Reviewers



# Components of WebKit



# Web Browsers

WebCore + JavaScriptCore

Safari



Eclair



WebCore + V8

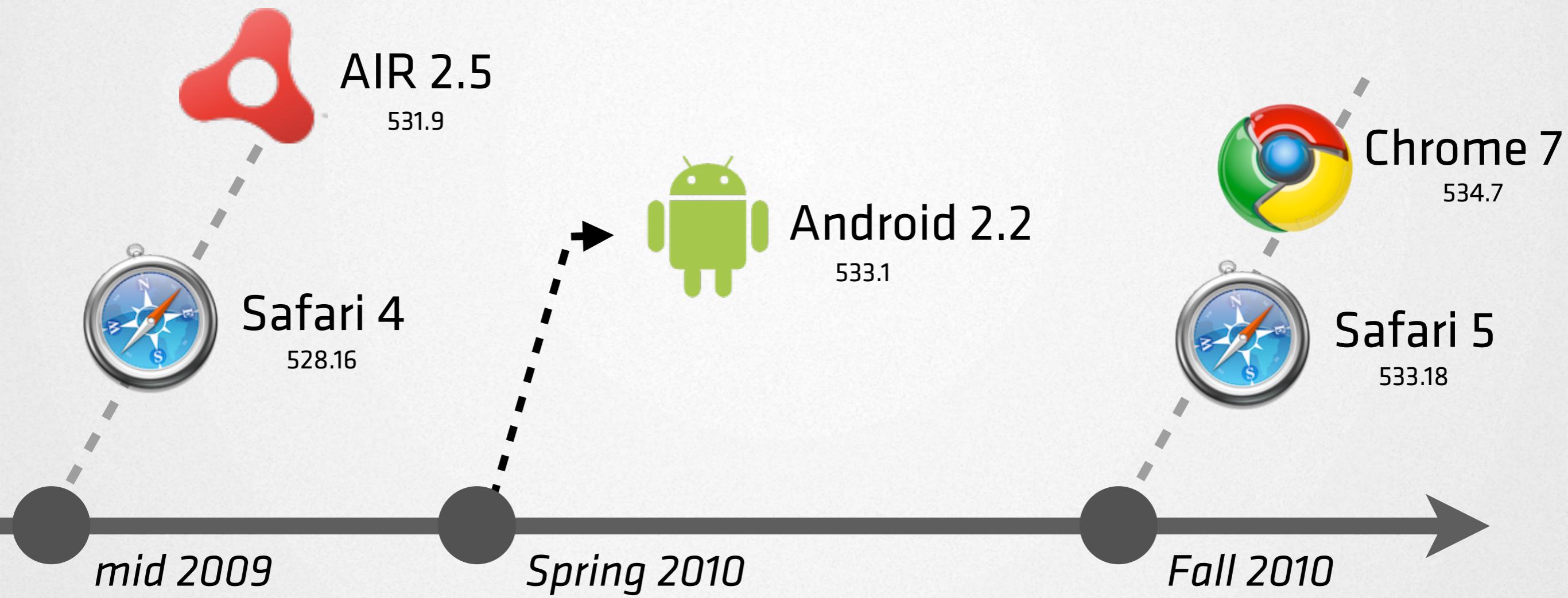
Chrome



Froyo



# How Fresh?



# Platform Abstractions

Network	Unicode	Clipboard
Graphics	Theme	Events
Thread	Geolocation	Timer



# Different “Ports”

WebCore  
graphics

GraphicsContext

Mac

Chromium

Qt

Gtk

CoreGraphics

Skia

QPainter

Cairo

graphics stack

# Get + Compile



# Requirements

- Subversion or Git
- C++ compiler
- Perl
- Python
- Various SDK



# Using Subversion

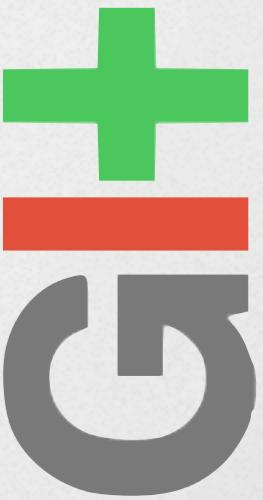
```
svn checkout http://svn.webkit.org/repository/
webkit/trunk webkit
cd webkit
```



# Using git

```
git clone git://git.webkit.org/WebKit.git  
cd WebKit
```

≈ 1.2 GB .git



# Build

`WebKitTools/Scripts/build-webkit`

**--qt** for Qt, **--gtk** for Gtk+  
**--debug** for “Debug” mode



# Launch

`WebKitTools/Scripts/run-launcher`

**--qt** for Qt, **--gtk** for Gtk+  
**--debug** for “Debug” mode



# Render Tree

paragraph

```
<html><body><p>Hello SenchaCon!</p></body></html>
```

```
layer at (0,0) size 800x600
```

```
  RenderView at (0,0) size 800x600
```

```
layer at (0,0) size 800x600
```

```
  RenderBlock {HTML} at (0,0) size 800x600
```

```
    RenderBody {BODY} at (8,8) size 784x576
```

```
      RenderBlock {P} at (0,0) size 784x18
```

```
        RenderText {#text} at (0,1) size 117x16
```

```
          text run at (0,1) width 117: "Hello SenchaCon!"
```



# Let's Start the PARTY



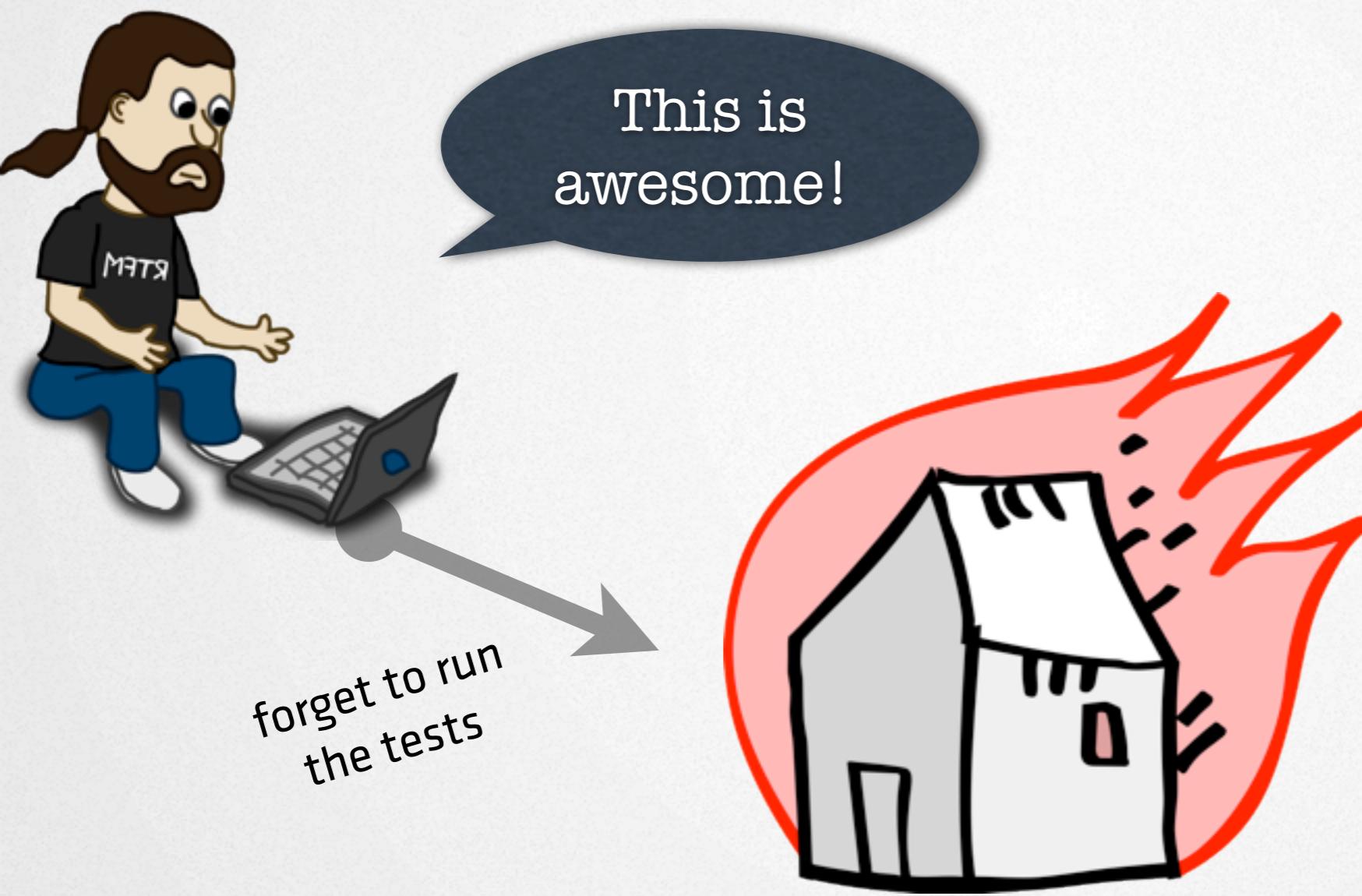
# Let's Start the PARTY



# Typical Scenario

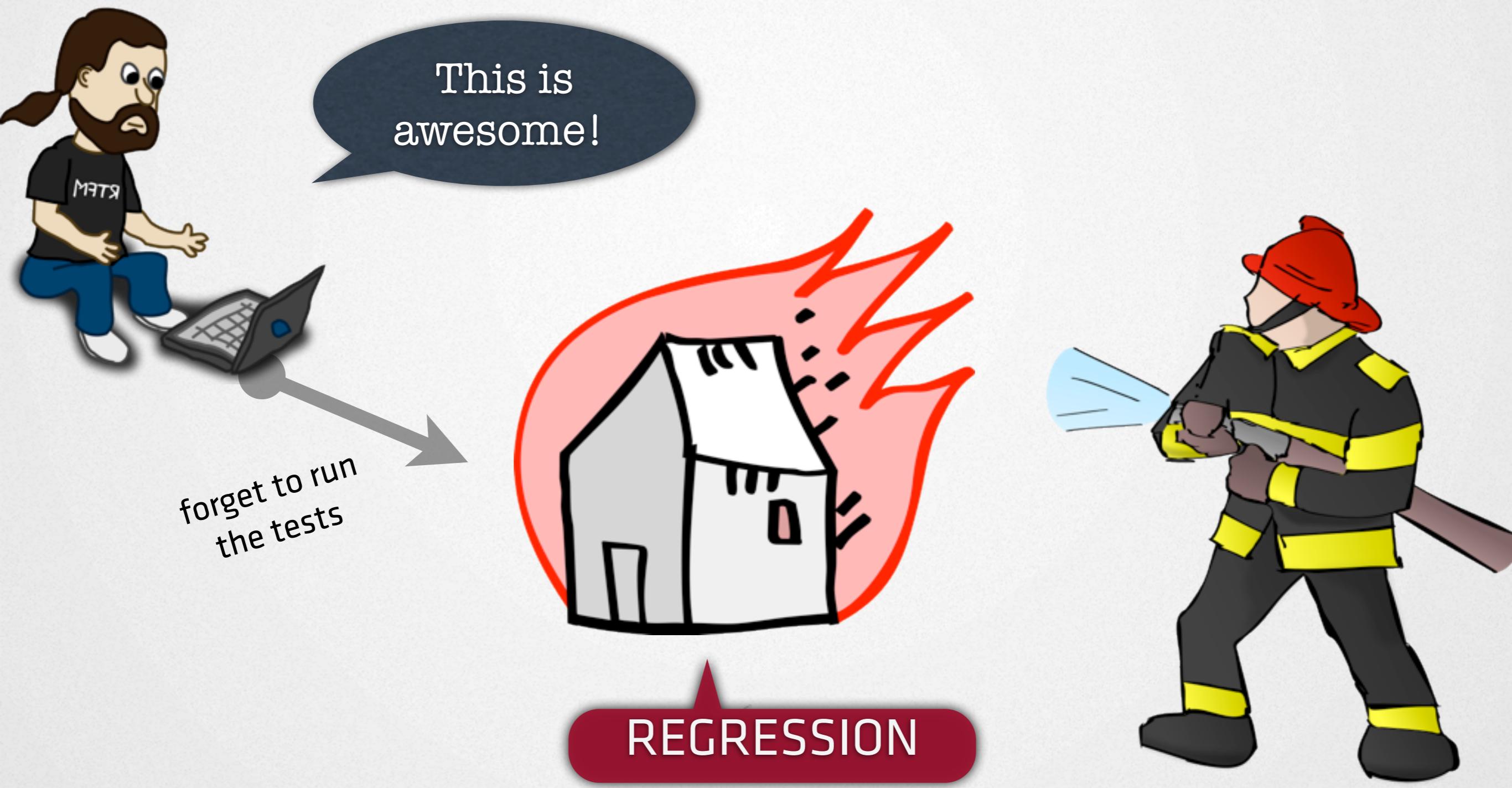


# Typical Scenario



REGRESSION

# Typical Scenario



# Using WebView

```
QWebView webView;  
webView.show();  
webView.setUrl(QUrl("http://www.sencha.com"));
```



# Capture to Image

```
QWebPage page;
QImage image(size, QImage::Format_ARGB32_Premultiplied);
image.fill(Qt::transparent);
QPainter p(&image);
page.mainFrame()->render(&p);
p.end();
image.save(fileName);
```



# Network Log

```
28: GET http://www.google.com/m/gp
292: Response 200 application/xhtml+xml; charset=UTF-8 0 bytes http://www.google.com/m/gp
311: GET data:image/gif;base64,R0lGODlhiA...
312: GET data:image/gif;base64,R0lGODlhJA...
312: GET data:image/gif;base64,R0lGODlhGA...
312: Response 0 image/gif 3611 bytes data:image/gif;base64,R0lGODlhiA...
312: Finish fail data:image/gif;base64,R0lGODlhiA...
312: Response 0 image/gif 284 bytes data:image/gif;base64,R0lGODlhJA...
312: Finish fail data:image/gif;base64,R0lGODlhJA...
312: Response 0 image/gif 178 bytes data:image/gif;base64,R0lGODlhGA...
312: Finish fail data:image/gif;base64,R0lGODlhGA...
317: Response 200 application/xhtml+xml; charset=UTF-8 0 bytes http://www.google.com/m/gp
324: Finish fail http://www.google.com/m/gp
328: GET http://www.google.com/m/gn/user?...
329: Finish success http://www.google.com/m/gn/user?...
```



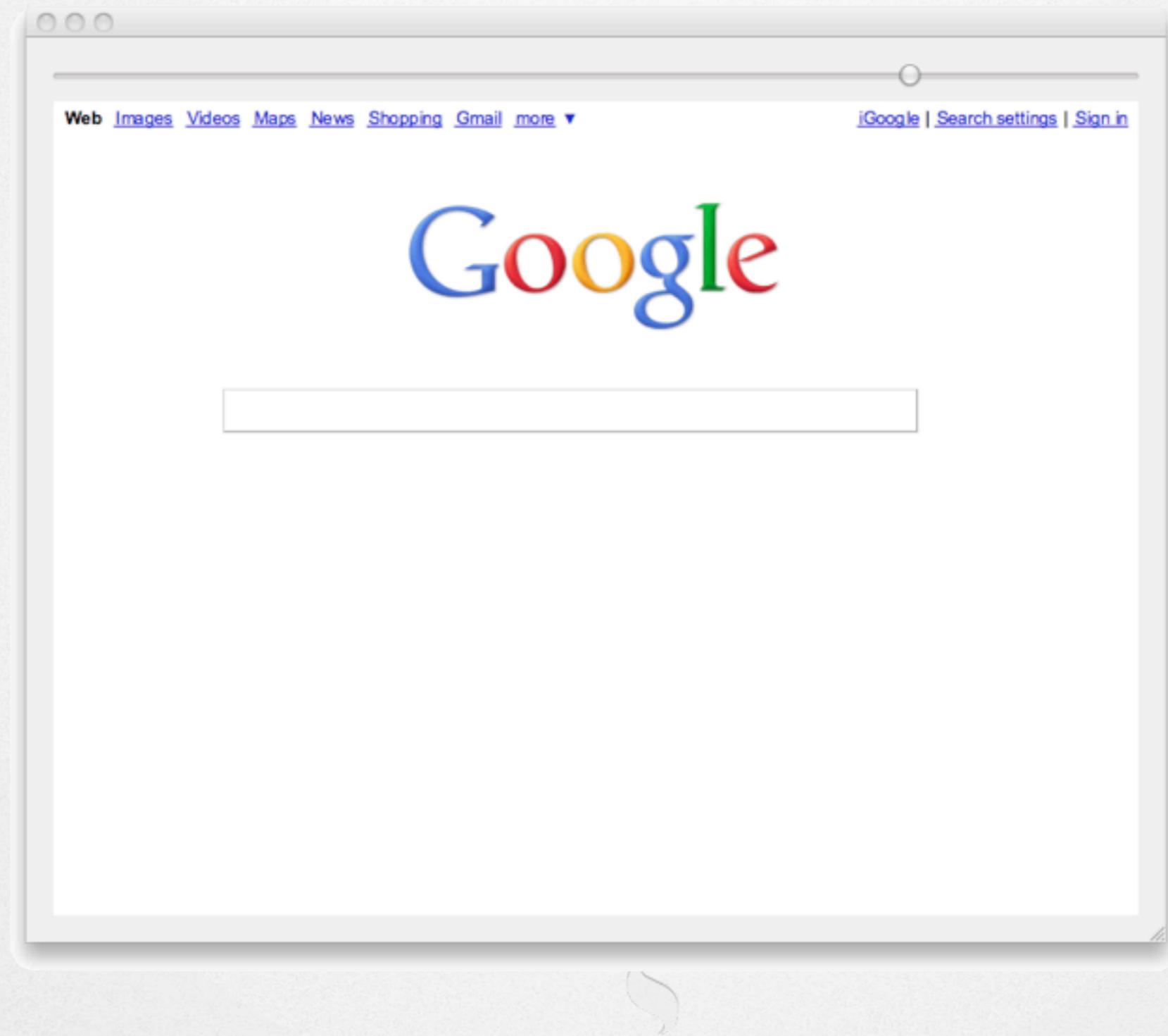
# Test Runner

```
specrunner SpecRunner.html  
5 specs, 0 failures in 0.013s
```

```
specrunner SpecRunner.html  
FAIL: 5 specs, 1 failure in 0.014s
```



# Recorder & Replayer



# Remote Console for



The image shows two views of the RemoteJS application. On the left, a desktop screenshot of the 'RemoteJS – Remote JavaScript Console for Android' window is displayed. It shows a command-line interface with the following JavaScript code entered and executed:

```
> console.log(document.title)
Map

> console.log(Ext.version)
0.9.8

> console.log(Ext.button)
undefined

> console.log(Ext.Button)
function () {h.apply(this,arguments);}

> console.log(Math.PI*3)
9.42477796076938

> Ext.getOrientation()

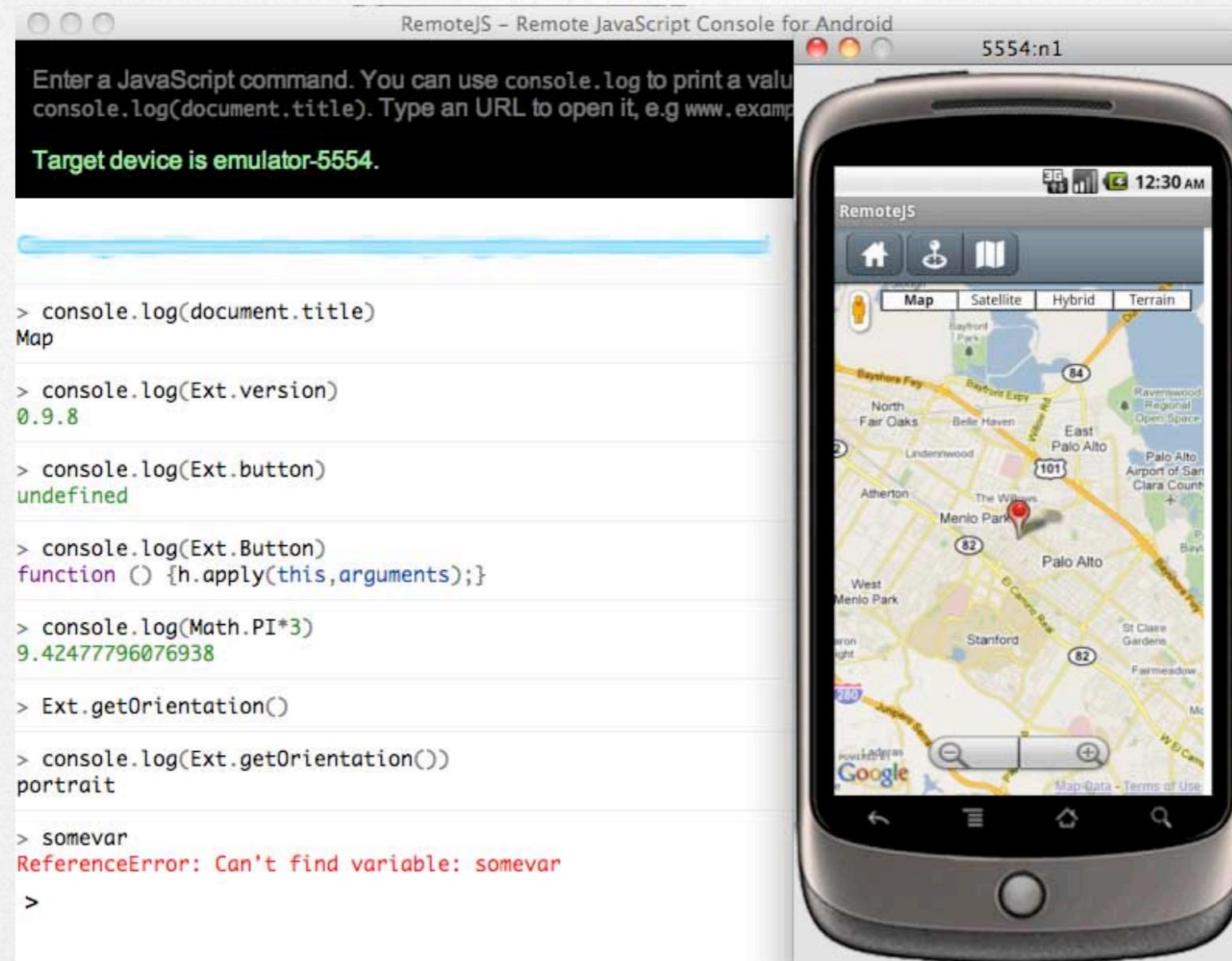
> console.log(Ext.getOrientation())
portrait

> somevar
ReferenceError: Can't find variable: somevar
>
```

On the right, a screenshot of an Android smartphone displays a map application titled 'RemoteJS'. The map shows the area around Menlo Park, Palo Alto, and Stanford, with various roads and landmarks labeled. A red marker is placed on the map near Menlo Park.

<http://github.com/senchalabs/android-tools>

# Remote Console for



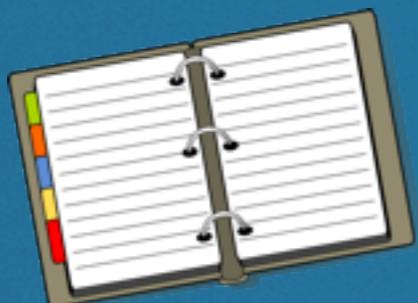
<http://github.com/senchalabs/android-tools>



# THANK YOU!



ariya @ sencha.com



ariya.blogspot.com



ariyahidayat

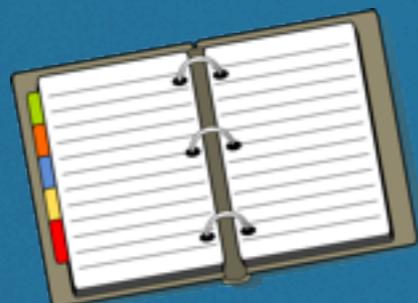


# THANK YOU!

We're hiring!



[ariya @ sencha.com](mailto:ariya@sencha.com)



[ariya.blogspot.com](http://ariya.blogspot.com)



[@ariyahidayat](https://twitter.com/ariyahidayat)

