Dave Rogers, Senior iOS Software Developer / Architect

408-234-2285 (C) ● drogers@cemico.com

Goal: Currently seeking a Senior iOS position, local to Silicon Valley, within a close-knit team, working for a great company with great product(s), who pride themselves on hiring capable smart people.

Mission*: "I have much experience and a solid background, yet there are many things still to learn. As a class valedictorian, I learn new things quickly, and am able to apply my background of quality programming to new environments. I am continually learning, and enjoy mentoring those around me. I am very reliable, a solid team member, and a strong independent producer."*

I love what I do, iOS, iOS, and more iOS ;) My passion began as a hobby in 2008/2009, did the startup venture in 2010, have pushed out 40'ish apps since, from idea to submission, and all steps in between. My iOS path began on Mac OSX Leopard with XCode 3 developing on my iPhone 3 under iOS 3, and has been very active every release since, including evaluating Swift when 1.x hit, and adopting it with full active development after v2.0 was released.

Prior to my iOS days, have much experience around Windows and Web applications using Microsoft Visual Studio predominantly with WebStorm as a companion. Windows applications were mainly in C/C++ with some in C#, and many utilizing MFC. C# recently crossed back into iOS applications with a few years of using Xamarin, both under MSVC and Xamarin Mac Studio. The web apps used ASP.NET and JavaScript, and spoke to a variety of databases, such as Oracle, Informix, MS SQL Server, etc.

Resume: http://cemico.com/resume.pdf   Cover: http://cemico.com/cover.pdf,   LinkedIn: http://www.linkedin.com/in/drogers

Apple Proficiencies (June 2008 – present)

- Swift v1-4, ObjC, XCode v3-9, iOS v3-11, C#, C++, C
- Full SDLC: Design, Code, Test, Submit, Agile
- App Store: ~40 apps, 100+ updates, 150k+ downloads

- Fastlane, Test Flight, Jenkins, Hockey App, OTA
- GA, Crashlytics, Xamarin, Parse, Flurry, NewRelic, Fabric
- REST, JSON, Instruments, APNS, MEAN, GIT, SVN, Jir

Primary Work Experience

Apr 2016 – Current, Onkore – Gamifying non-sporting events, starting with reality TV shows. We've released 4 App Store products, with multiple apps having multiple skins / themes, weekly+ updates. Our 3rd season / skinning of The Bachelor is the current app, although all the apps share a common code base re-written from scratch using Swift 2, 3 and now 4. The architecture is a Mongo database, Node.js backend, REST and Socket.IO, sitting behind AWS and Atlas. Many debugging utilities, profiling and optimizations, custom controls, animations, thread and message management, auto navigation from APN and Branch, responsible for the architecture as well as most the coding of the iOS application. Heavy theme and appearance work. Responsible for all aspects of the app, from conception to submission.
Note: some app(s) temporarily taken down due to seasons ending
Apps: https://itunes.apple.com/us/developer/onkore/id965498769

Mar 2014 – Apr 2016, ALLDATA / AutoZone – Multi-billion dollar company, automotive application which integrates a hybrid interaction between the user interface (web) and a WiFi / BLE device plugged in to the car. Many complexities around the communications between the two, the massaging of the data, background vs. foreground processing, and the interaction between the native app and the hybrid JS interface. Direct questioning from the CTO after my first week why I thought it best to toss 99% of the past 2-3 years of development and re-write it. 4 months later, submitted to App Store, reducing the code footprint from ~2m to under 40k, while making it more robust and maintainable. Since the initial release, have updated it, converted it to C# / Xamarin, to 64-bit, and to iOS 8 and iOS 9 compatibility, as well as enhancing with barcode / vin scanning. App communicates with a REST backend. Note: also working on iPhone utility app, but currently unreleased. Final form of the app is C#, HTML 5, Angular.js, Bootstrap, Typescript, Node.js, while communicating with the hardware via WiFi or Bluetooth.
App: https://itunes.apple.com/us/app/alldata-mobile/id697105390?mt=8.

Feb 2013 – Mar 2014, Twrrl – The app started in iOS 5, then to 6, and finally to 8. This app I created from scratch, and turned in to an outstanding video application, exercising many aspects of technology, from low level system integration to user interaction in sharing the resulting video to any number of share targets.   App communicates with a REST backend. Note: app pulled in 2016 when company closed (have working copy on device)

May 2012 – Feb 2013, Computer Associates – Enterprise app shown at CA World 2013. The app was an enterprise solution for securing emails and attachments, both decrypt and encrypt, using REST, animations, custom buttons, mimic native mail app.

Apr 2010 – Dec 2012, The Creative App Co – Few dozen apps submitted. This is where I cut my teeth. Starting with iOS 3, I did my learning while producing a few demo apps. Finding some direction, and a partner in a startup, I began working to create an app which could easily be skinned for a new client, a true morphing app. Many things went in to this with the results being at runtime, everything was dynamic, the content, the screens, how it looked, etc. I learned much about the intricacies of the app submission process during this time. I also enhanced it with a large metrics layer which tracked usage, and pushed it to an online server where I wrote the code to receive the data store it into a MS SQL database, and also wrote the web interface to view all the metrics. I also had automatic scheduling of daily emails showing various metrics activity. In addition, I wrote the complete backend APNS, also with MS SQL integration, and a web front end to schedule pushes. This was a complete client / server solution with no external tools (remember, iOS 3 days didn't have many free options available at this time). The biggest success was a Justin Bieber Fan Club app which had 70k+ downloads, and a Rihanna Fan Club app which had 50k+ downloads. There was also a Drinking Rulebook app which was hitting ~2.5k downloads a month.

Jun 2008 – Apr 2010, Cemico Inc – iOS Development beginning from iOS3 / XCode 3. Develop many prototype / test apps to learn various areas of iOS app development. This was an excellent way to build solid foundations, allowing self-paced focus to fully understand the many areas comprising an iOS application.

Background Work Experience

May 2017 – Jul 2017, AccuTimer – Slow side project developing a video timer app with embedded per-frame video stream enhancements, watermark overlay, filters, and text overlays.

Oct 2016 – Jan 2017, iGo.Live – Similar to Periscope app, a Broadcast / Viewer application. Brought in to fix the issues which local engineers could not find. Rewrote their socket code, updated to Swift, full Alert / Window class for complex alert system, general difficult bugs and enhancements, version 2.0 release.
App: https://itunes.apple.com/us/app/igo.live/id1146243047?mt=8

Nov 2014 – Mar 2016, Onkore – An early exploration of concept which turned in to an ObjC / Node / Sockets / REST iPad demo app which allowed the means to host and run games on any live event, focusing on non-sporting events. Complex technology which constantly crashed before I started.

Aug 2015 – Jan 2016, Crimep – A map application which provided crime incidents, notifications, geo-location fencing, custom map enhancements, etc. App never released. In addition to the iPhone client, I also wrote the REST backend interface in node.js / mongodb for development, which was used as the template for production.

May 2015 – Aug 2015, Twrrl – Upgrade of app (see Twrrl work below) from iOS 6 to iOS 8. Many routines were replaced and/or deprecated, upgrade was big effort.

Feb 2014 – Aug 2015, 401k Day Trading – Released Mar-2016. I designed and implemented 95% of this app, from the logic, the frameworks, the live quotes, account tracking, options, etc. The last 5% (in-app purchase feature and final touches) was added by another. This is a companion app to a 401k Trading book found on Amazon.
 App: https://itunes.apple.com/us/app/rebalance-your-401k/id1083731354?mt=8

Apr 2015 – May 2015, Skully – worked alongside 2 other developers, taking early POC work and turning it into an actual product. I had two main areas of responsibilities. First, I was involved with handling all the contact management, social sharing, camera integration, photo caching, custom album, collection viewing, deleting, and sharing. Secondly, I was involved with the map functionality, incorporating Skobbler maps, with a custom overlay interface, turn-by-turn directions, communicating the map data not only visually, but also via audio and BLE to the helmet. This is another cool product. The client hired in-house talent at about 90% completion. My favorite quote from the project PM when he asked me to turn my focus from the social sharing to the map SDK: *"You seem to be really really good at digging into 3rd party frameworks and making shit work….",* April 28, 2015 via MB. Note: company closed in 2016 due to fraud.

Sep 2014 – Dec 2014, United Nations / AT &T – unreleased, did initial POC work. This is by far the biggest name of any app I've done – it's the non-profit United Nations World Food Programme, funded by AT & T. The app involved being able to donate for various fund raising campaigns throughout the world. In addition to doing the iOS app, I also supplied the talent, and managed the Android app, the CMS web interface, and the backend, node.js and mongodb. Separately releaseed as:
App: https://itunes.apple.com/us/app/sharethemeal/id977130010?mt=8

Oct 2014 – Nov 2014, Feb 2013 – Aug 2013, Computer Associates – 2 separate CA contracts. The first enterprise app was shown at CA World 2014 and was loosely an enterprise version of DropBox (this one got $500k funding from the prototype), using OAUTH and more normal backend REST services mimicking the DropBox API. The second, was a training tool used by the salesforce to demonstrate a subset of features available, in helping to drive revenue of said products. This app also used REST services, and included a means to run in offline mode for ease of testing and demonstrating the product. Each product delivered more functionality in less projected time and dollars.

May 2014 - Aug 2014, Jan 2014 – Feb 2014, Auction Mobility – I completed two physical contracts, each time helping out where demands were great (and declining a third time due to availability). This app was an older, established and dated, iOS app, using many things which has since evolved. The code base was meant as a generic solution to produce auction software for many different clients – i.e., a single code base serving many clients. Utilizing my direct experience in this type of app, I came in and re-organized the code, setting it on a path for better maintainability and quicker release cycles and app re-skins. The app has many intricate and cool features, which I touched in my time helping out in numerous customers releases. This app used one of the best backend REST services I've seen. Most of the apps are enterprise, but the ones I submitted can be seen at:
App: https://itunes.apple.com/us/app/hart-davis-hart-wine-auctions/id890633292?mt=8
App: https://itunes.apple.com/us/app/pages/id787524467

Education and Merits

San Jose State University, 3.9 GPA
Mount Pleasant High School, Valedictorian
Spotlight on local TV for elite student and athlete
First Varsity tennis player
Black belt in Tae Kwon Do with dozen+ years teaching
Numerous video course, such as: Stanford CS193p iOS course for iOS8, iOS9, iOS10
Onkore DWTS app recognized by appPicker as #1 Hot App on April 3, 2017, http://cemico.com/apppicker-dwts-04032017.png
Five apps shown in Las Vegas SEMA and CA World conferences

2-3 Day Interview "coding example projects"
- Sep-2017, Weather App, Swift 3 / Xcode 8, modelled after the native Weather app, task was to pull weather data online, use AlamoFire, collection view, animations.
  https://github.com/cemico/Weather
- Sep-2017, Alarm App, Swift 4 / Xcode 9 GM, loosely following the alarm portion of the native clock app, task was to be able to schedule alerts, receive them in background, foreground, not running, display customized images on the alert.
  https://github.com/cemico/PeelAlarm