

# Aleksander Balicki

I'm a **Lead iOS Developer** with a little under 3 years professional experience. I've done complex front-end (**CALayer** **anchorPoint** and **transform** fun), back-end (parallel **NSOperationQueue** processing) and build system tasks (full command line automation of App Store submission).

I use **AppCode** instead of Xcode, because it makes coding less mundane. I prefer to use **CocoaPods**, which became a necessity with the number of Apps in Wikia. I always have **Dash** (documentation browser) installed on all my iDevices. I often read Apple manuals at home for fun.

Phone: +48 695 198 977  
Email: [balicki.aleksander@gmail.com](mailto:balicki.aleksander@gmail.com)

Blog: <http://alitra.ghost.io/>  
Github: <http://github.com/alitra/>

## Employment

[Wikia](#), 2013-2015, **iOS Team Tech Lead**, 1 year 4 months, current.

- Developing features and bug fixes for [Game Guides](#) and [Community Apps](#)
- Second iOS developer in the company, big impact on the architecture and design of the apps
- Working on scaling the environment and the code to handle hundreds of [Community Apps](#)
- Setup of the build process of Wikia iOS apps: **Jenkins**, **Testflight**, **JIRA API**, **Handsfree Automation**
- Developing a modularization framework for iOS and implementing it in Wikia apps
- Implementing **Analytics**, **i18n**, **Local and Push Notifications** and **A/B Tests** in the apps

[MorrisCooke](#), 2012-2013, **Lead iOS developer**, 11 months.

- Lead developer of [StickAround](#), team of 4 people
- Implemented a parallel, responsive engine for path crossing detection on iOS
- Enhanced the presentation-to-video compression time by a factor of 2 in [Explain Everything](#)
- Working with a large codebase (500k lines of code)
- Advanced and complex **UIView** and **CALayer** management and **animations**
- Teaching junior developers about the platform and good software design
- Introduced a proper bug tracker in the company, code style guidelines.
- Architecture of the slide grid in [Final Argument](#)
- Ported Organizer+ to the new **iPhone 5 resolution** and **iOS 6**

Timepie (never released), 2012, **AppleScript developer**, 1 month

[Intres](#), 2012, **iOS and OS X developer** 10 months.

- StoreQ - Web-based file hosting (Dropbox-like) service
  - Created and designed the client apps for both iOS and OS X
  - Frameworks used: **fmdb**, **Sparkle**, **CDEvents**, **Growl**
- Integrated **iAd** and **AdMob** into [dGuide](#)
- Optimized the amount of data [dGuide](#) downloads upon startup (from 20MB down to 100KB)

## Education

Computer Science, University of Wrocław, Master's Degree, 2014.

Computer Science, University of Wrocław, Bachelor's Degree, 2011.

## Skills

### iOS Development

**Front end** I know how to create multitouch **UIKit** animations that don't snap, when you add or remove fingers. I have the power to control the time (just in **Core Animation** objects implementing **CAMediaTiming**). I do all the layout in code instead of .xib files, as it's safer and easier to code review. Recently I started using **AutoLayout**, but most of my complex layout work was done by hand in layoutSubviews. I always pester the designers, that something is not according to the **Apple Human Interface Guidelines**.

**Back end** I don't understand why people use **AFNetworking** if they don't cancel the **NSOperations**, they could have just used the Apple libraries. I learned the hard way, that besides using a **Core Data** managed object context on only one queue, you also have to initialize it on the same queue. During code reviews I usually have to explain what a **dispatch\_group** does and that doing a lot of boolean flags isn't the way. I wanted to add **Swift** to the app, but **CocoaPods** not yet support Swift fully.

**Testing** I've written unit tests, but most of my testing was done in **Frank**, selenium-like ruby acceptance test framework. I wrote support for resetting the state of the simulator and helped the QA team add accessibility-Labels. I've helped create a Frank scenario to generate all the required screenshots in the Apps, because doing 5 screens on 4 devices for 12 languages for 25 apps isn't feasible manually.

**Build system** Building 125+ apps every time you develop something is easier when you have full automation. I developed a build system that works from the command line, supports multiple Apple programs (including **Enterprise**), signs the IPA properly and uploads to **Testflight** or the **iTunes Connect**. I use **xctool** as the reporters nice and easily produce **oclint** required files. Some scripts required **AppleScript**, as not everything has a command-line equivalent.

Languages - Objective-C, Haskell, Python, Ruby, C, able to learn new programming languages fast.

Knowledge of algorithms, data structures, databases and networking.

Linux/Unix command line tools (vim, git) and scripting, network and Linux administration.

Contributed small patches to open source projects: Stash, xxxterm, jumanji, rust;  
haskell libraries: alex, imgurder, derive, Safe, XMonad.Util.Paste

Interested in compilers, static code analysis, development tools, OS X, iOS and Linux.

## Projects

imgur-directory-listing - a photoblog, that uploads the pictures to [imgur](#), designed for home servers with a slow upload connections, initially written in **Ruby on Rails**, rewritten in **Yesod**.