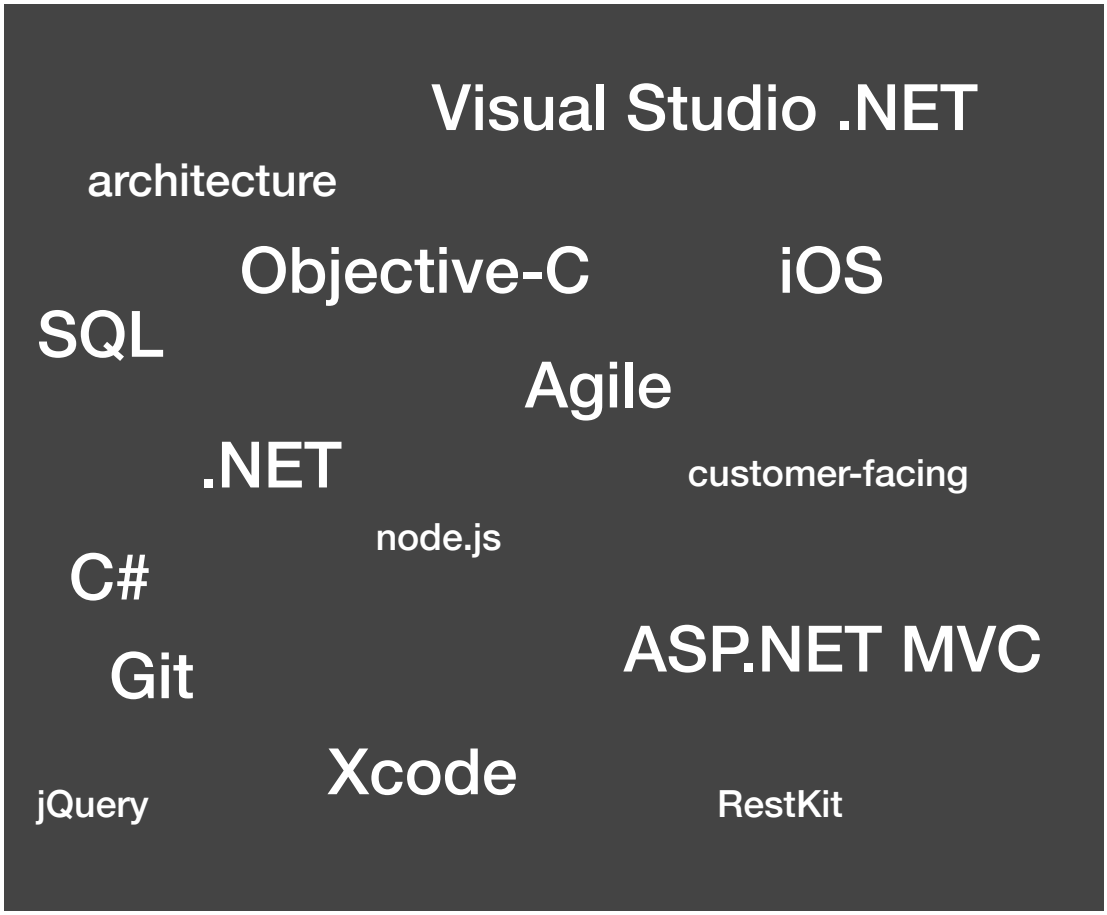
SCOTT McKENZIE

April 2013



ANZ
National Bank
Westpac
Alphero
NZRU
Paper Kite

summary

This is a portfolio of several iOS apps I have worked on. My experience goes much further than that with experience as a software developer and consultant with Trade Me, Gen-i, Datacom and Omron.

I have 15 years of software development experience, mostly on the Microsoft platform. I started with Visual C++ for building visual neural networks. I used the same skills on SCADA systems. I moved onto C# for a variety of consulting roles in New Zealand. At Trade Me I worked on the primary website with a focus of performance and optimisation. Most projects I have worked on have used SQL Server.

I have always invested time and effort in core skills for building software and this approach has seen me pick up new skills quickly and succeed on challenging projects many times.

Over the last few years I have started diversifying technical skills to cover non-Microsoft technologies and products; iOS is an integral part of this. This has exposed to me to many other tools that are less applicable to the Microsoft platform.

I enjoy all the challenges this industry offers; delivery, leadership, technology, customer satisfaction, innovation and on-going learning. Each project is different.

my style
development

My development and leadership styles are quite similar. I like to keep things simple, stay agile and responsive, so I can meet requirements quickly. I tend to design and plan my way to avoid "big-bang" deliveries, preferring an incremental approach with an element of fail-fast.

iOS development
what I use

AFNetworking - when low-level comms is required
RestKit - for REST (and similar) service integration
GHTestRunner - for developing and running unit tests. I find it offers more granular control than the built-in OCUnit.
OCMock - a mocking framework
Test Flight - for delivering applications to testers and customers. This provides similar behaviour to that of the App Store
Urban Airship - for providing Apple Push Notifications
Google Analytics - for observing how people use the app
Lumberjack - for high performance and granular logging. It's better than NSLog.
Calabash - for BDD based automated testing
Git - for source control
Gitolite - for Git access control
Source Tree - a Git IDE for OSX/ Windows which supports GitFlow
Xcode - for all iOS development
Magical Record - for SQLite integration
ARC - for memory management
Core Data - for in-app persistence and in-memory management
Jenkins - for continuous integration and, if possible, deployment
Model-View-ViewModel pattern - a great pattern from Microsoft that I implement in Obj-C using categories

core skills matrix

C#	2002-2013	Advanced
Obj-C	2009-2013	Advanced
Git	2010-2013	Good
SQL	2004-2013	Good
OOA/D	1995-2013	Advanced
ASP.NET/MVC	2004-2013	Advanced
Agile	2008-2013	Advanced
Integration	2004-2013	Good
JavaScript	2004-2013	Good
Patterns	1998-2013	Advanced

learning
what I'm looking into

Gamification - I'm attending the online Coursera course.
node.js - this is a great choice of service for mobile apps and has the potential to allow for more client-server code sharing
ThoughtWorks technology radar - to help keep up with a fast paced industry
Android development - this is dominating the market in numbers so I just have to learn it

Scott McKenzie
scottmckenzie@outlook.com
@trustyfrog.com
nz.linkedin.com/in/mckenzie/

a bit about me

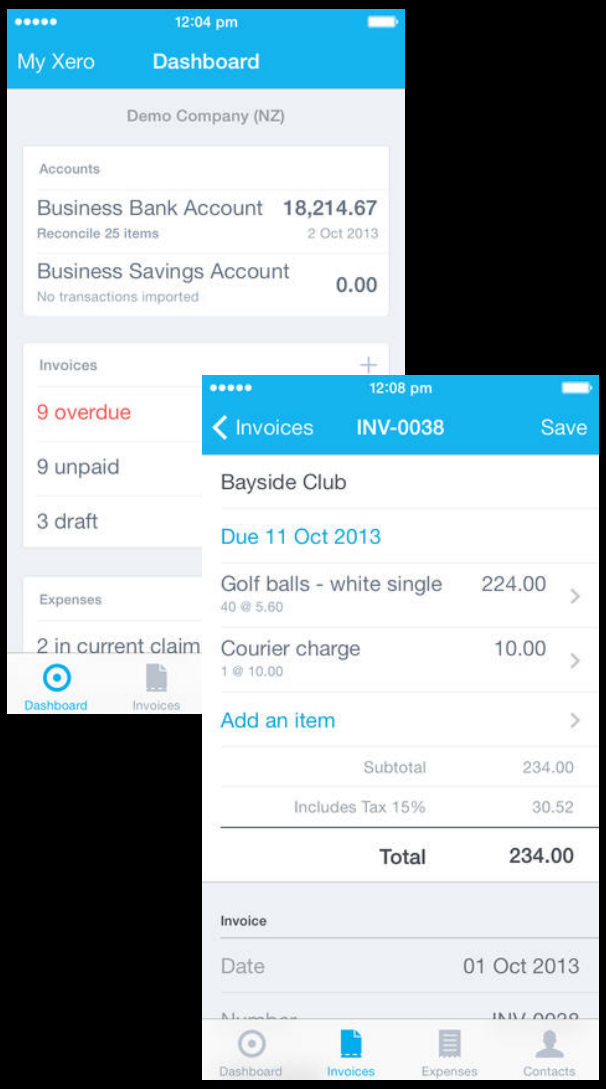
I'm fortunate in that I'm passionate about my day job. Relaxing for me is coding and keeping up to date with technology.

I'm learning the violin; a challenge as I have little natural musical ability

I love reading and writing. This is mainly science-fiction and fantasy, but I'll pick up anything interesting.

I run and swim frequently. I perform bodyweight exercises to stay fit.

I like trying new sports so over the last few years I have snowboarded, surfed and windsurfed.



Manage your business on the go straight from your iPhone or Android device. Login with a 4-digit passcode then view your accounts, create and send invoices, chase up outstanding invoices, snap a picture of receipts, submit expense claims and more.

http://www.xero.com/nz/accounting-software/mobile/

- Git
- Xcode 5
- REST API and JSON integration
- Objective-C
- iOS 6, 7
- Sqlite3
- AFNetworking
- HockeyApp
- Confluence / Jira

Xero Touch iPhone app

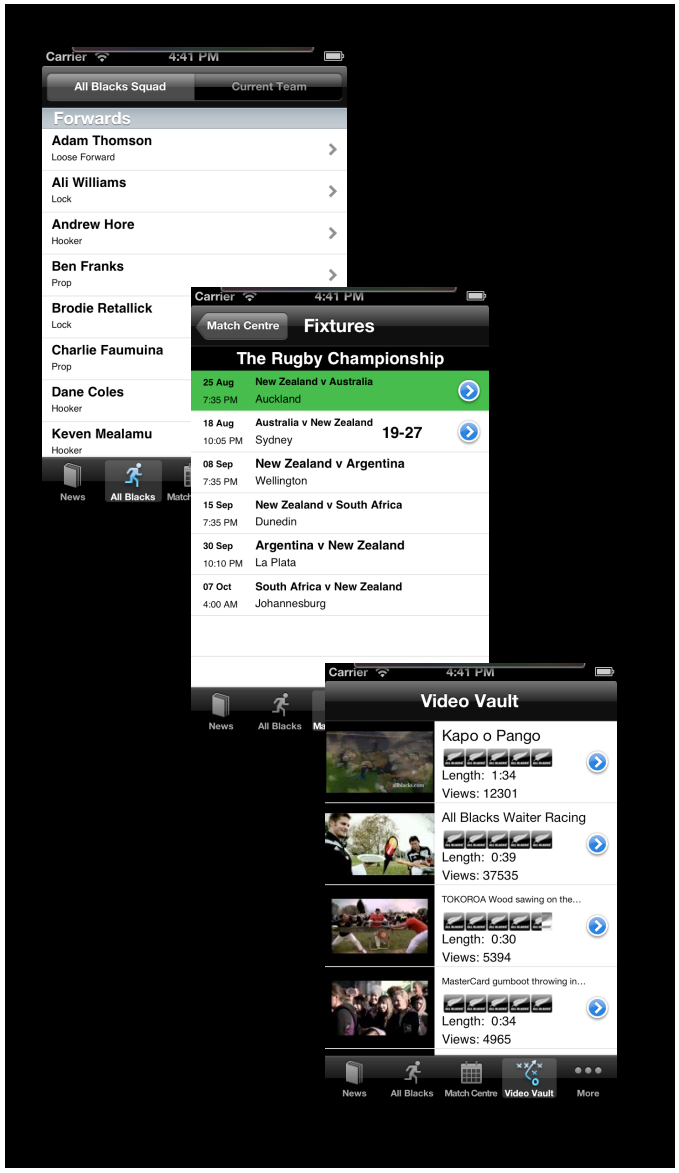
Xero is the emerging leader in online accounting software. Being part of an exceptional team building an app that is used globally is a great opportunity. And there's much more to come.

Working with exceptional people is the best way to improve. And that has definitely been the case for me on this team.

As part of a small team it has been possible to touch all areas of functionality on this app; API integration, security, bank reconciliation, user interface, invoices, expenses, contacts, database, future features and support for iOS 7.

Working on a global app represents challenges not usually encountered in the other apps I have worked on. This requires additional attention to detail and rigour to maintain the quality of app that is expected.

Some of the requirements have meant the use of a non standard Sqlite3 implementation to support R-Tree for geospatial support and Full Text Search (FTS3).



The Official All Blacks app was commissioned by the NZRU to promote the All Blacks brand

http://allblacks.com

- Subversion
- Xcode 3
- XML service integration
- Objective-C
- NSXmlParser
- Caching
- Asynchronous image loading
- iOS 3
- YouTube service integration

NZRU Official All Blacks iPhone app

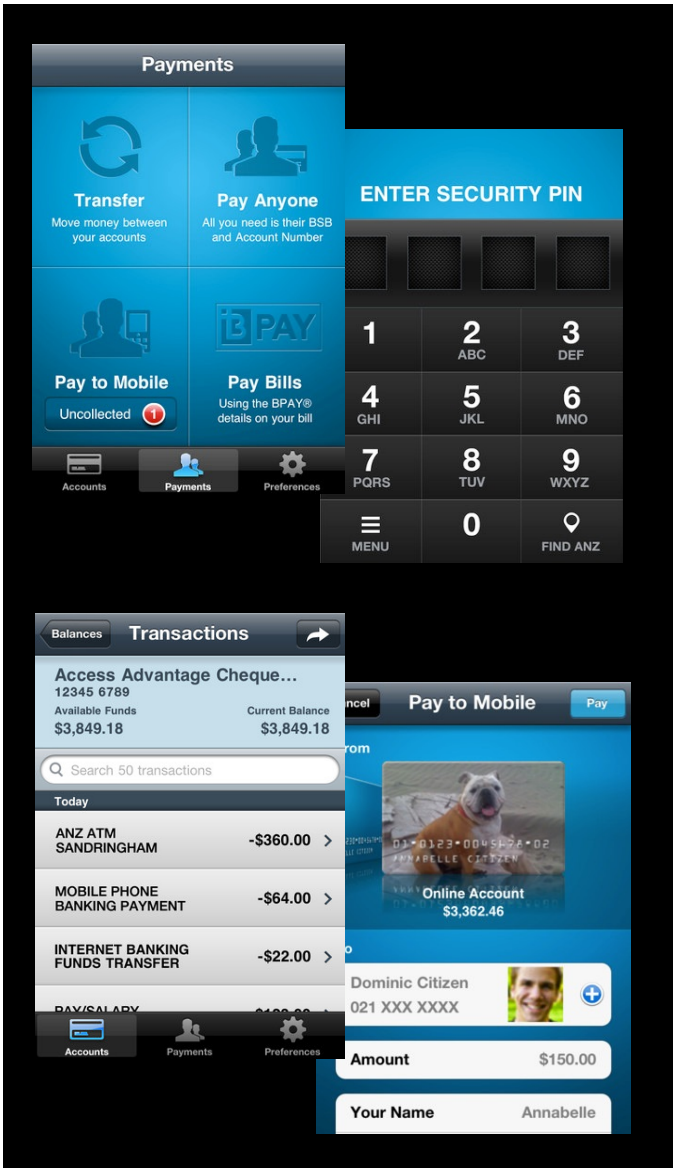
The start to my iOS history was a lucky one. I set myself the goal of learning a new language and decided on Objective-C in the context of iPhone development. After declaring this over social media a manager at ANZ approached me to build the All Blacks for his start-up, Paper Kite.

I was able to produce a prototype in a short period of time to validate capability and then followed up with the final app.

I set the requirements for the XML feeds for the much of the data for news, players and statistics. Later I encouraged the use of You Tube's API to deliver video into the app.

I integrated with XML based feeds, cached data locally to keep bandwidth usage low, loading images for news and players asynchronously.

As my first app I pleased that it was accepted first-time. I carefully monitored errors logs for many months afterwards and no crash reports were ever submitted.



ANZ goMoney is a secure banking iOS app in the hands of over a million users.

http://www.gomoney.anz.com

- Subversion and Git
- Xcode 3, 4
- GSOAP
- OCUnit
- GHTestRunner
- Hudson, Jenkins
- BDD
- Agile
- XCConfig
- Core Graphics
- Core Animation
- SQLite
- Objective-C
- Team leadership

ANZ goMoney

Following on from a strong reputation of delivery in ANZ and my work on the All Blacks app I was asked to join the goMoney team. A high profile app for the bank.

I worked on Pay Anyone and Pay to Mobile features for both AU and NZ variants of goMoney. This work touched most of the mobile banking stack; service integration using GSOAP, address book integration, business logic, validation, controller logic and UI.

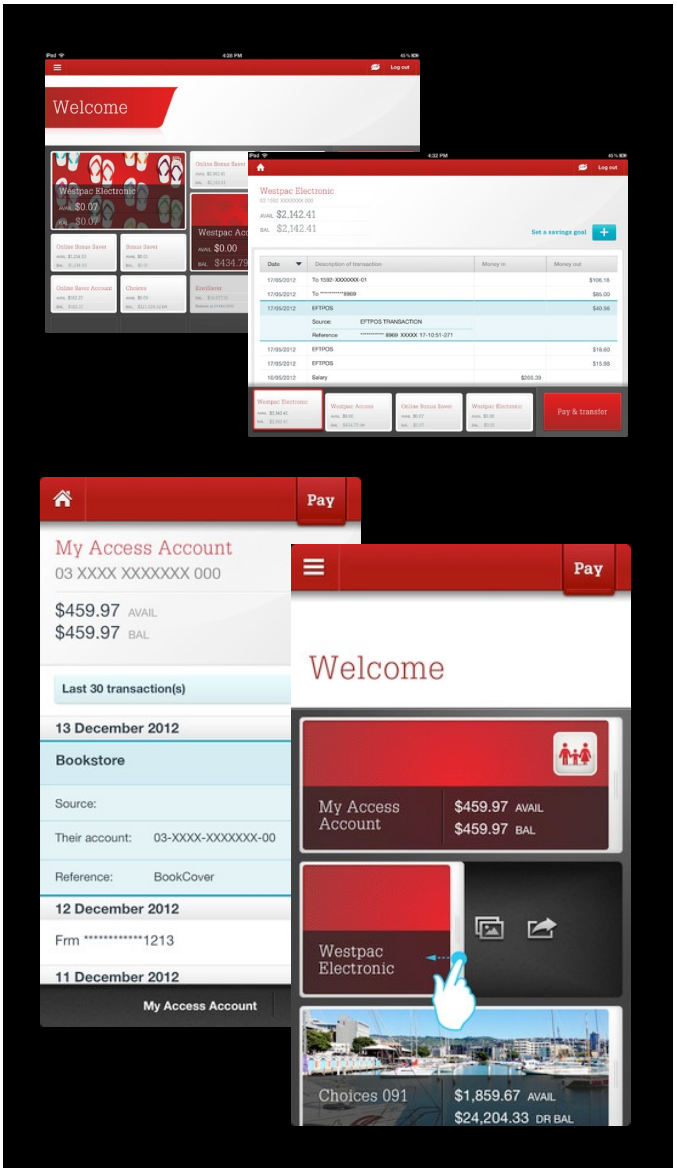
I worked on other releases including support for NZ Simplification (ANZ's core merger) and maintenance.

In an effort to improve code quality I increased the compiler warnings monitored, fixed them and configured the project to treat warnings as errors.

I worked on prototypes for international money transfers and for iPad.

I chaired and ran the ANZ Mobile Design Committee; a group trying to align technical thinking, share ideas between teams and reach decisions on how to progress.

I led the NZ and AU development teams for several months, conducting code reviews, influencing design, liaising with architects and the business



I am part of the Alphero team responsible for developing the very successful and stylish iPad and iPhone banking apps for Westpac

http://www.westpac.co.nz/branch-mobile-online/while-you-re-mobile/

- Git/Gitolite
- SmartGit
- Xcode 4
- RestKit
- OCUnit
- Jenkins
- Agile using Kanban
- Core Graphics
- Core Animation
- Objective-C
- iPad
- iPhone
- Customer-facing

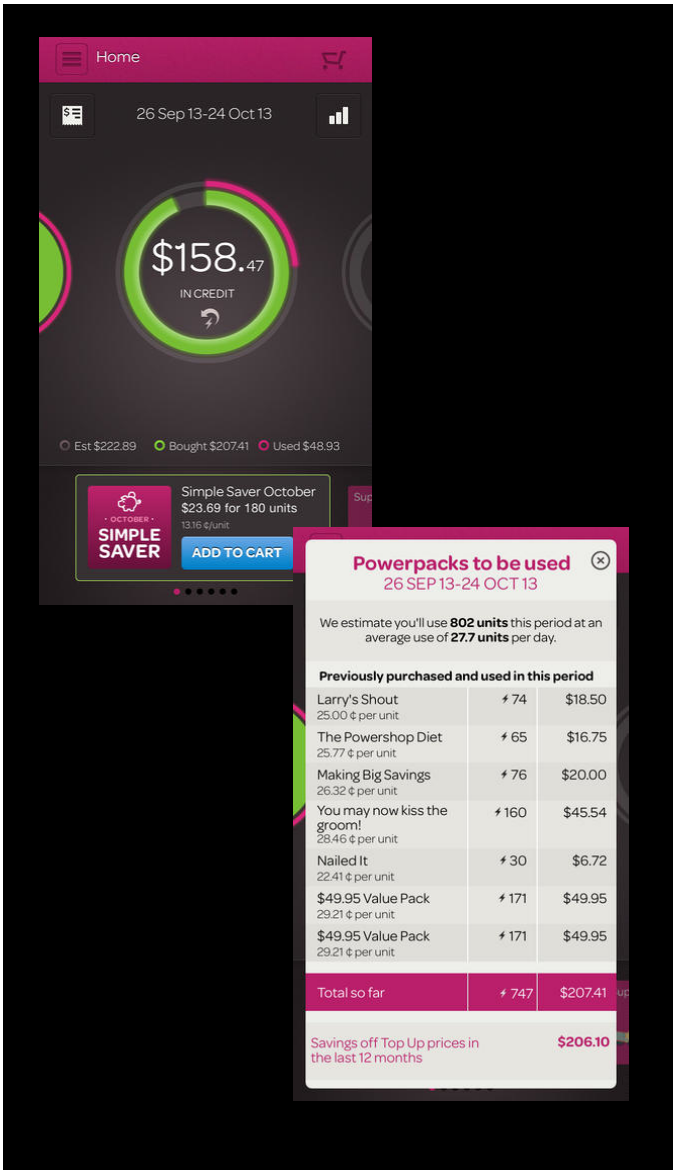
Westpac iPad and iPhone Banking app

I worked on Westpac's iPad and iPhone banking apps for Alphero, a mobile consultancy. The app is visually sophisticated with complex transitions. It has been very popular with Westpac's customers and saw a significant climb in the ratings.

I worked on the dashboard and account transaction screens. This covered service integration, data representations, business logic and the UI. Probably the most challenging aspect was getting the animations between tables and scroll views

I'm pleased with a decorator pattern I implemented over the model object allowing multiple transaction details to be requested simultaneously by a user. This pattern neatly tied the UI with the service layer and the underlying model.

I also worked on a secure persistence layer for storing personalisation. This held up to the rigour of external security testing.



I am part of the Alphero team responsible for developing the iPhone Powershop

http://www.powershop.co.nz/mobile-apps/

- Git
- Source Tree with Git Flow
- Xcode 4
- RestKit
- Jenkins
- Kanban
- Core Graphics
- Core Animation
- Objective-C
- iPhone
- Calabash automated testing
- APNS
- Trello

Powershop iPhone app

This is a customer facing iPhone app for purchasing products with some great animations, transitions and interactions.

I worked with the RestKit framework on REST integration. RestKit makes web service integration much simpler but I did have to code some complex dynamic mappings.

I worked on some subtle, grouped animations for purchases.

An exciting aspect of this project is the use of Calabash for automated testing of the iOS app. For each story I write automation tests in a BDD style.

As part of this project we used Git Flow workflow for using Source Tree.