NICO

WEB DEVELOPMENT TIMELINE

Improved process and use of tooling. For example I started ensuring my code is well commented, using **version control** (Git), tracking tasks (Trello), automated build steps (Grunt), **unit testing** (Jasmine), etc.

Took over development of a complex ~30k LoC Laravel+Magento code base near the end of a project. Orientated myself with minimal documentation or contact with the previous developer, and made the required adjustments to PDF rendering logic.

Working with the DevOps lead, **provisioned and configured AWS resources using CloudFormation,** and set up a CI pipeline using Jenkins.
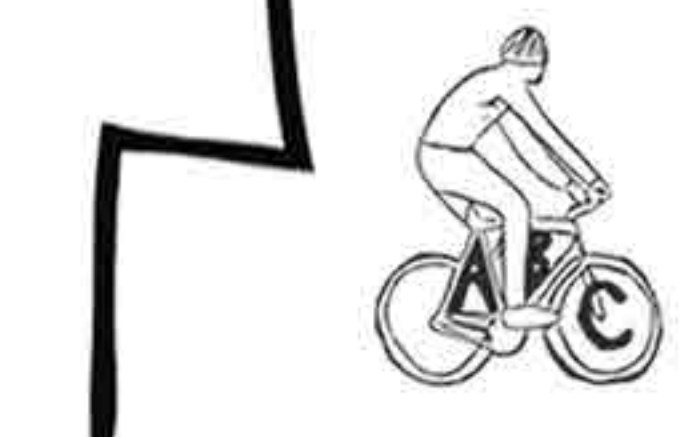
THE FUTURE

Work on large scale systems with real scaling challenges

Lead projects and/or a small team

Get more involved in the UX / product design aspects of creating systems

Started following industry media (Smashing Magazine, A List Apart, HTML5/JavaScript Weekly, Mozilla Hacks, V8 Blog, Hacker News, etc).

Graduated from the University of Bristol with a 2:1 in Maths & Philosophy.

Started learning about common software design patterns and **changed the emphasis in my work from just making things work, to discovering best practices and trying to do things The Right Way.**

Architected and built a complete analytics collection and querying system. including data ingestion in **Rust**, query APIs in **Python** and a dashboard in **Angular 2**. Data was stored in **Amazon Redshift**.

First paid web development work. Involved writing CSS for old and buggy browsers.

| School / Freelance work | Oxford | Mubaloo | Bristol | IdeaSq | G+V | Toumetis | Somo |

2005   2010   2012   2014   2016   2018

AlwaysBeCycling.org.uk: Involved negotiating and **working with the client**, creating the **visual design** from scratch (over several iterations), and coding a **custom PHP CMS and booking system**.

Created an offline-first responsive single-page mobile web app. Used backbone.js + lodash + jQuery. It is now used by thousands of people a day.

Gave talk on "Functional Programming in JavaScript" at CodeHub's 1-day workshop.

Worked on an MDM product. **Coordinated development** between the different developers, and **took responsibility for the functioning of the overall solution.**

Various coding experiments. **Experience led to more reliable code and less time spent chasing errors.**

Gained experience working in a team. We worked in an agile fashion presenting our work to the client every 2 weeks. Git was used to coordinate development.
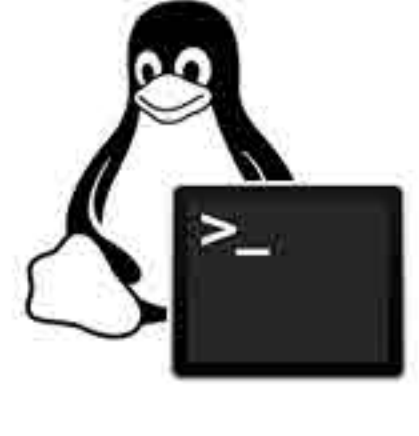
Worked remotely, created a new **Laravel** based architecture for an existing PHP website. Learnt a ton about **configuring a live production server** using UNIX tools over SSH.

Learnt Rust, and through it many concepts of **low-level programming** (e.g. stack vs heap, RAII, allocation) **functional programming** (e.g. Algebraic Data Types and Traits) and some (like ownership) that are unique to Rust.

STRENGTHS & KEY SKILLS

Independent problem solving. I am good at solving new and unfamiliar problems independently, I am not afraid to attempt things I have not done before, and I am quick to pick up new skills.

Programming ability & experience. Along with a good conceptual understanding of programming language features, I have extensive 'real-world' experience creating complex applications and the deep knowledge of practical issues and platform quirks that comes from having done something before.

Technical communication. I am good at communicating technical issues to non-technical people. And conversely, I am good at translating 'business requirements' into technical requirements.

Interest in UX/product issues. Whether it is a GUI or an API, I care that the systems that I am creating work well for their consumers. I think it is important to have someone looking out for this on the dev team who can catch the little issues as they arise.

A collaborative approach. We all have different areas of expertise, strengths, weaknesses and perspectives. Let's work together.

Technology selection. I keep up with the latest and greatest technologies (and their reputations from real users), and I am good at selecting technologies that are appropriate for a given project.

Technical Architecture & Systems Design I can help with defining the overall approach to a business problem, choosing appropriate technologies and designing a cohesive set of systems which work well together.

Helping & mentoring others. A chore for some, but a joy for me. It's one of my favourite parts of the job.

Friendliness. As well as the products I am working on, I try to care for the people I am working with.

Don't let this list limit you. I am good at learning new skills quickly and successfully implementing projects with previously unfamilar technologies.

AREAS OF EXPERTISE

Languages
JavaScript / TypeScript / Python / PHP / Rust

Frontend Ecosystem
React / Redux / Angular 2+ / TypeScript / RxJS / Webpack / Babel / ES2018 / CSS & Layout / SASS / PostCSS / Service Workers / Cordova

Backend
Node.js / Express / PHP / Laravel / Python / REST / MySQL / PostgreSQL / Redis / Nginx

AWS
S3 / EC2 / Route 53 / RDS / SQS / Redshift / +more

DevOps / Other
Docker / Git / Github / Bash & Unix CLI / SSH / Jenkins / Gitlab CI

And I have a little experience with...
Go / Java / Objective-C / Swift / C++ / Ruby

TECH FAVOURITES

PostgreSQL / Redis Both performant, featureful, and incredibly reliable. My goto in their relevant niches.

React + TypeScript An abstraction that really works, with a modern, flexible type system. Makes code simpler and more reliable. What's not to like?

Rust C++ level performance, Haskell-like "correct if it compiles", and ergonomic like JavaScript or Python. An under-appreciated gem.

TECH I'M WATCHING

React Native / Flutter Cross-platform apps are a big deal. Would use RN now, hopeful for Flutter soon.

Kubernetes Probably only suitable for large systems, but reviews seem to be almost universally positive.

C# / .NET Always a well designed language. Now open sourced and linux compatible it's looking very promsing.

10k+ top 4% overall

"The app has been a huge success and has become a global benchmark for HP by seeing this rolled out across EMEA and beyond."
Daniel Kiaie, Account Manager, HP

FIND ME ONLINE

nicoburns.com
stackoverflow.com/u/140293
github.com/nicoburns

EDUCATION

2016	University of Bristol	BSc. Maths & Philosophy (2:1)
2012	University of Oxford	BA. PPE (unfinished)
2011	A-levels	A*AAB

CONTACT

Nico Burns
nico@nicoburns.com
07773 667 110

Please see page 2 for details of my work history and other experience

This CV looks great printed on A4 (double-sided, flip on long-side)

Senior Full-Stack Developer (Contract), Somo
Sep 2017 – Aug 2018

Somo are a digital agency based in Bristol. For the duration of my time at Somo, I was working on a Mobile Device Management product for a Somo's client, a new startup. The app was technically challenging, pushing the limits of what is possible on contemporary mobile operating systems, and won a DADI award for Technical Innovation.

The Drum DADI Awards

Admin Interface (React/Redux)
Created an admin interface for configuring the product. Nothing fancy, just the usual React stack: Webpack, Babel, React, Redux, React Router, React Form, SASS, PostCSS, etc.

Admin APIs and Mobile App APIs (PHP / Laravel)
Mostly simple CRUD APIs making use of Laravel's many helpful productivity features. A few complex queries where we fell back to raw SQL. The app-facing APIs were load-balanced to deal with a high traffic levels. I would have loved to have a go at really pushing the optimisation here! Health-checks were implemented to alert us if the live system stopped running correctly.

Apple MDM Server (PHP / Laravel)
Implemented the relevant parts of the MDM server protocol from scratch by reading the spec/documentation provided by Apple.

Super-User Mode (Server-rendered PHP)
Per my suggestion, the super-user mode used server-rendered PHP. This allowed for extremely rapid feature development of a non user-facing part of the system where design was less important.

Dev and Testing Environments (Docker)
Created a uniform development environment using docker compose. This was also leveraged to provide on-demand environments for running automated tests.

Whole-System Design and Integration
Constraints imposed by the OS meant that implementing the specified features required complex interaction between client apps, server-side components and OS MDM features. Part of my role in the project was to ensure that the components worked correctly together as a whole system, and not just individually.

CI and DevOps (Jenkins + AWS)
I helped configure a jenkins pipeline that automatically built services and ran tests and deployed them to a dev environment. I also helped manage our AWS Elastic Beanstalk based dev and production environments.

Testing (PHPUnit) and Reliability
The backend APIs (including the MDM server) were ~90% covered by API-level tests. This proved to be a good balance: catching most bugs while not requiring excessive time spent on writing tests. The system was praised by the client for it's reliability, with only 1 minor issue being found in production.

Distributed Agile Development Process
The team for this app consisted of designers, developers, QA testers, a product owner, and scrum master. This team was distributed over 3 offices coordinated over Slack and video conferences. The process was iterative with 2 week sprints with feedback after each.

Full-Stack Developer, Toumetis
Jun 2016 – July 2017

Toumetis are a small digital agency based in Bristol. Half of their business is focused hybrid mobile apps, and the other half is making an analytics product. In my time with the company I experienced both halves of the business.

System Design / Architecture
Completely redesigned an existing analytics system:
- Replaced custom in-memory database with AWS Redshift.
- Seperated monolithic application into separate ingestion and querying microservices.
- Fixed severe performance (30s --> ~500ms) and reliability issues.

Web Dashboards (TypeScript/Angular2+)
Angular 2 based web dashboard frontend for Toumetis's analytics product "Cascadence".
- Frontend built on Angular2, TypeScript, and RxJS.
- Built using Webpack. Styles managed using SASS and CSS modules.
- Unit tested using Karma and Jasmine, and e2e tested using Protractor under dockerized headless chrome.
- Charts rendered using highcharts.

Backend Query APIs (Python/AioHttp + Redshift)
RESTful APIs to allow querying of analytics data to support the dashboards described above.
- Backend REST APIs written in Python using AioHttp.
- Amazon Redshift used to allow efficient OLAP queries of large data sets.
- Build and test automated through Gitlab CI. Built Docker containers deployed to AWS.

High-performance ETL pipeline (Rust + Amazon S3)
A data pipeline to process analytics data from a proprietory log format into a CSVs for loading into a Redshift data warehouse.
- Built in the Rust language. Allowing C++ level performance without the security/safety risks that come with C++.
- Stores all state in S3 to allow individual instances to be treated as stateless, and restarted without disrupting processing jobs.
- Pluggable IO connectors allow switching between batch and online processing modes.
- Incremental processing avoids the need to reprocess large batches of data.
- Memory efficient, running with consistently less than 150mb RAM despite processing 100s of gigabytes of data.
- HIghly reliable with uptime measured in months.

Hybrid App for Education Startup (TypeScript/Node/AngularJS)
A hybrid web app for an educational startup.
- Frontend in Angular 1 and an in-house hybrid wrapper
- Backend APIs created with NodeJS with TypeScript with PostgreSQL as the database

Internal Build/Dev Tool (Node.js + React)
A new web-based tool to modernise a legacy QT in-house build tool. Allowed developers on Toucan based apps to use the devtools from the latest version of Chrome (or browser of their choice) rather than buggy devtools from an ancient version of QTWebView.

Note: Between September 2013 and May 2016 I completed a 3-year degree in Maths & Philosophy. I only worked during summer holidays during this period

Web Developer, IdeaSquares
Jun – Sep 2014

Hired for the summer period to refactor an unstructured and increasingly difficult to maintain code base, and to update the functionality of the website to meet new business requirements.

- Ported existing "spaghetti code" to the Laravel framework.
- Implemented new front-end designs.
- Carried out general website maintenance and updates
- Advised on development plan and technology choices.
- Worked remotely, communicating regularly via email and skype.

PHP Developer, G & V
Jun – Sep 2015

Brought in to finish off a project after their previous developer left the company, I worked at Goram & Vincent for 2 months in their office in Bristol.

- Took over a large (30k LoC), well-written but complex codebase.
- Implemented PDF rendering logic. Updated Magento integration.
- Setup a staging environment and automated deployment.
- Added documentation and prepared for the next developer.
- Advised on technical issues for upcoming projects.

Mobile Web Developer, Mubaloo
### Feb – Sep 2013

My first proper job, I worked as part of Mubaloo's Web Team for 8 months before starting my degree. Mubaloo are an award winning agency who specialise in iOS, Android and Mobile Web apps.

Responsibilities & Achievements
- Developing web apps using modern web technologies including HTML5, CSS3, SASS, JavaScript, jQuery, and Backbone.
- Gained experience of the browsers (and WebViews) of on all major mobile platforms including iOS, Android, and Windows Phone.
- Participated in client meetings, helping to interpret requirements from and explain technical decisions to our generally non-technical points of contact.
- Helped out with scoping and time estimation, UX design, and device testing.
- Pushed for and helped facilitate a company wide transition from SVN to Git.

Top Value App
This app gives sales people in the field an easy way to find product and pricing information. When developing this app, we worked in sprints, obtaining feedback on our current build while we worked on the next one. This allowed us to really streamline the UX, and build an app which works as it's users expect it to. It is now used by thousands of people each day.
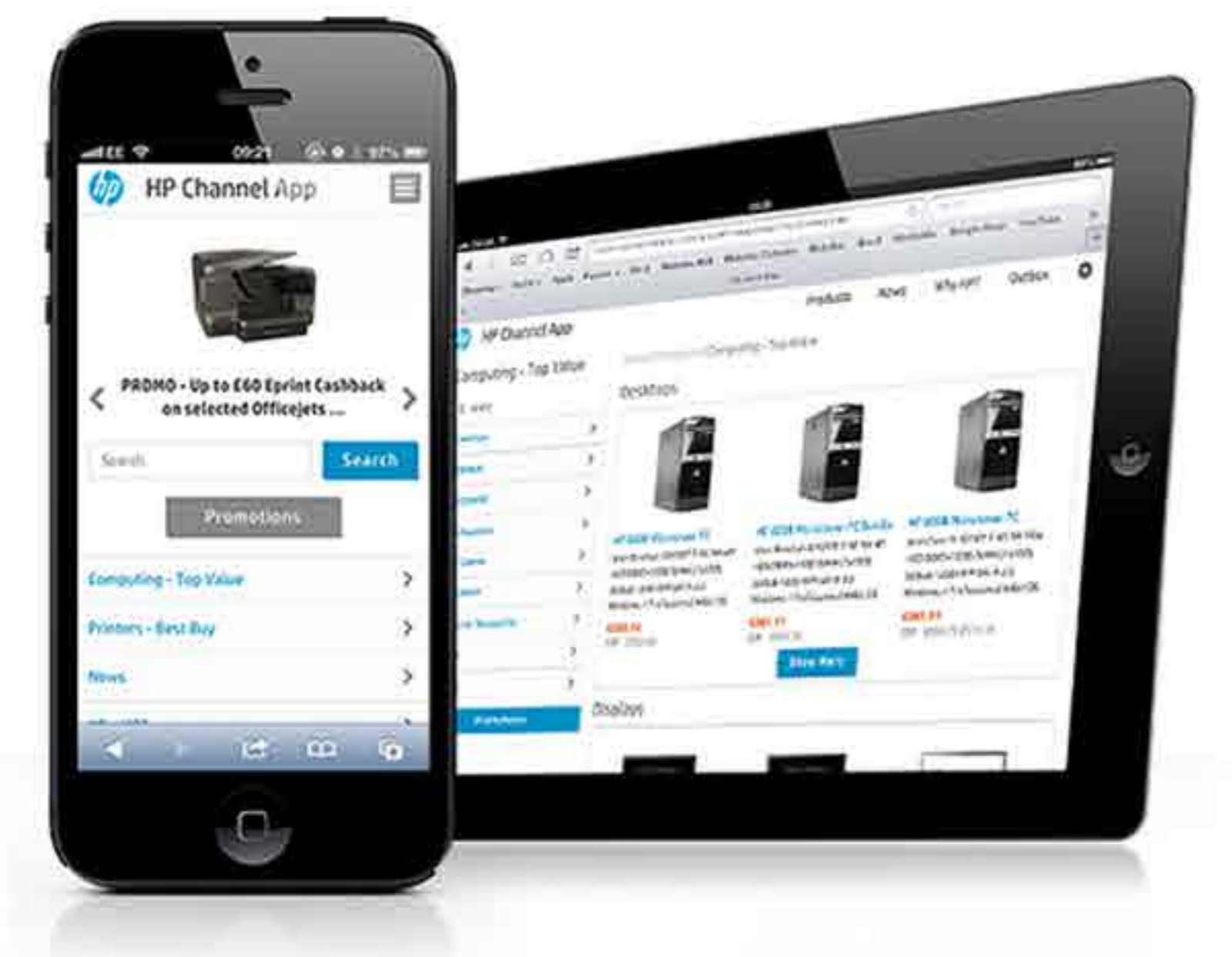
"The app has been a huge success and has become a global benchmark for HP by seeing this rolled out across EMEA and beyond."
Daniel Kiaie, Account Manager

GhostEdit WYSIWYG Editor, Personal Project
2008 - 2010 https://github.com/nicoburns/ghostedit

GhostEdit is a Javascript based WYSIWYG editor which came out of my frustration at the poor usability of the more well known editors like TinyMCE. It took interface cues from Microsoft's Ribbon UI, which is both simple to navigate, and familiar to users. In addition to this, it tries to make editing...

...actions intuitive and discoverable (e.g. placing the remove link button right next to the link), and to do as much as possible automatically (e.g. if no text is selected when 'bold' is pressed, then current word will be bolded). GhostEdit was particularly challenging from a technical point of view, due to the incredibly buggy nature of browsers' contenteditable implementations. This necessitatedthe overiding of browser default actions when the 'enter' and 'backspace' keys are pressed, among other things. Adding to the complexity was the fact that older versions of IE have a completely different range implementation.

Other Early Work
2006 - 2011

2011 Always Be Cycling (CMS w/booking system)
An informational website and booking system for a local charity. I was responsible for the whole solution including liasing with the client, visual design, photography & content creation, and development. The website was based on a custom PHP CMS with article editing functionality and a course booking module. It worked with modern browsers and IE6+.

2009 Tony Coll (static website)
My first paid development work. Involved content updates to a static HTML website, fixing the CSS to work with non-IE browsers, and factoring out common page elements using simple PHP includes.

2006 Fantasy Football Website
My first complex project: a PHP + MySQL based fantasy football website, complete with user registration, the ability to select a tea, and an admin area for updating player scores. Code quality was awful and there were security holes everywhere, but it was fully-featured, and was actually used by around 50-100 people.