

Sam Kington

33 McLaren Crescent, Glasgow G20 0LD, +44 796 816 3525
CPAN / GitHub / Codeberg : skington
sam@illuminated.co.uk | [@skington@glasgow.social](https://twitter.com/skington)

Software development principles: craft and care

I've been writing back-end software remotely for more than 25 years, using a variety of server environments and frameworks. I know how to manage my time, and not get stuck. I understand the virtues of laziness, impatience and hubris. I try to not write clever code.

I'm comfortable with test-driven development where appropriate (i.e. especially when fixing bugs or refactoring). As befits an Internet geek with an Arts degree, I like to *start* by writing documentation and code examples: it's a good way to work out what I'm trying to do, and whether I would want to use my own code.

I'm very comfortable with the debugger, and have tracked down problems colleagues couldn't.

Communication principles: make it easy for people to get involved

People read code far more often than they write it, so I take my time when naming things, and make sure my code is well-structured and -commented. I rewrite history so my git commits can be read individually (especially important when refactoring), and I try hard to split up large PRs into manageable chunks so reviewers have a fighting chance of understanding what's going on. I aim to write tests that spark adversarial participation, and I'm not afraid of whimsy.

Software should always be evolving, so I try not to design too much up-front. Sometimes it's OK to say "we don't know exactly how we want to do this yet", and for code to look like that.

I like talking to other people in the business about our technology and what's possible. I'm increasingly interested in using the various speeds of communication available to us to get the best results: when it's best to call someone and talk things through in real time, and when you're better off writing things down and letting people think about it at their own pace.

Places I've worked

Amtivo — February 2024 - April 2026

I joined some ex-Congenica colleagues, trying to understand and modernise an automated ISO 27001 etc. automated assessment codebase. I wrote comprehensive test libraries and got rid of a fair few performance pain points.

Congenica — January 2021 - January 2024

I took a crash course in precision genomics, and worked out how oncology was different from rare disease and how best to model that in the shared codebase. As effectively co-tech-lead of our team, I wrote minimal design documents, mentored new devs, code-reviewed all of their python code, and talked with Product and Clinical regularly so we could win a EU-wide tender.

Skills and buzzwords

Perl: DBIx::Class, Moose, Moo, Dancer, Catalyst, Mojolicious, Test::Class::Moose etc. etc.

Linux (Debian), Postgres, MySQL / MariaDB, git. Docker, Kubernetes, AWS S3. GitHub, CircleCI, Jenkins. Jira, Confluence, Redmine. Scrum, Kanban.

Also dabble in: Python, Swift, Javascript, HTML, CSS, C.

I got promoted to a higher-tier “talk to more people in the company” role, where I became one of the rotating build managers, and learned how we marshalled our hefty kubernetes-based pipelines. I took over organising the developer talks and gave a fair few, with titles such as “Test failures are your friend (Don’t be satisfied that your tests pass until you’ve seen them fail)”.

UK2 — January 2001 - December 2020

A dot-com boom domain registration and hosting company realised it needed a renewal system. I worked for years afterwards to maintain the systems and transition them from founder code to something approaching best practices.

Eventually we hired a proper development team to help write a replacement Modern Perl system, replacing CGI scripts with Javascript + REST, and we managed to avoid second-system syndrome.

DataCash — January 1998 - December 2000

I had to report line-item details of a dot-com boom company suddenly processing apparently everybody’s credit cards on the Internet. A virtuous circle happened where we hired smart people and they recommended their smart friends in turn, and I learned a lot about the craft of programming.

WWWDesign Ltd. — March 1997 - December 1997

I built or helped build a number of websites.

Glasgow University WebSoc — 1994- 1997 (student volunteer)

We set up a club whose sole purpose was to use a loophole and give students their own web page - basically geocities.gla.ac.uk.

Open Source

From time to time I’ve had an itch I needed to scratch. These are my favourite CPAN modules that I’ve written.

Data::Tersify – save your scrollbar from unexpected huge data structures.

LWP::JSON::Tiny – make LWP JSON-friendly.

Test::Mock::LWP::Distilled – use dev-friendly mocks in tests.

Timer::Milestones – profile your code without the issues of Devel::NYTProf.

Education

University of Glasgow — MA (Hons) Politics & Philosophy, 1996

Lycée Bertran de Born, Périgueux — Baccalauréat C (science), 1992

I’m therefore borderline bilingual in French.

Non-work stuff I like

Classically-trained pianist.

Table-top roleplaying (also my social group and source of mental health).

Politics and policy, current affairs.

Sometimes I decide I have things to say, so I have a blog. I try to not be Too Online, though.