Benjamin **Woods**

ben@bjqw.me

About me

Full-stack generalist, leader, manager, and **enemy of waste**, with experience in startups, big tech, and academia.

I work best as **glue**, bringing teams together and filling in gaps.

I take pride in **reducing waste**, prefer to adhere to lean and **Agile** processes, and believe in **documentation-first development**.

Areas of specialization

Python · Integration
Glue code · Bug hunting
Team optimization
Requirements analysis
Testing · Tooling

Technologies

Python || 6+ years
JS / TS || 3+ years

Python	● ● ● ●
pytest	● ● ● ●
Typescript	● ● ● ○
Node.js	● ● ● ○
Next.js	● ● ● ○
jest	● ● ● ○
flask	● ● ● ○
jwt	● ● ● ○
react	● ● ● ○
drf	● ● ● ○
numpy	● ● ● ○
cucumber	● ● ○ ○
django	● ● ○ ○
fastapi	● ● ○ ○
git	● ● ● ●
poetry	● ● ● ○
yarn	● ● ● ○
AWS	● ● ● ○
GCP	● ● ● ○
Docker	● ● ● ○
Terraform	● ● ● ○
Kubernetes	● ● ○ ○

Languages

English · native
German · conversational

benjamin-woods
benjaminwoods
benjaminwoods
soyapencil

Recent Experience

2024–now
0y 8m

Senior Software Developer
University of Cambridge · Cambridge, UK

I currently work in the UIS DevOps group at the University of Cambridge, who fulfill central university IT development and IT operations.

I work primarily as an independent contributor on a portfolio of products and services including the website for undergraduate admissions, internal systems for authorization and login, timetabling, and cookie compliance.

I work as a software generalist, completing both full-stack engineering across multiple teams as well as infrastructure and deployment in more of an SRE capacity. I give rich code review, and balance speed against priority in an organisation which has heavy engineering requirements and lean resourcing. The UIS DevOps portfolio is wide and heterogenous, requiring me to quickly learn new codebases, pave paths to quick and effective solutions, while championing for code improvement and good engineering practices. I follow TDD to produce high quality code fast.

Technologies and Frameworks

- **Planning + Design**: I use Mermaid to sketch user workflows and draw technical diagrams. I use Cucumber to contract feature designs, and use these to achieve alignment between stakeholders in cross-functional teams. I use GitLab to track work streams that stretch across multiple repos and products. The teams I work with follow traditional Agile as well as derivatives such as Kanban and Scrum.
- **TypeScript**: I use a variety of different frontend and server-side JavaScript and TypeScript frameworks with the Node.js runtime, including React, Gatsby, Axios, Firebase SDK, and Next.js. Most of time is spent working on critical features, as well as adding deep mocking to enable richer tests using Jest that better follow engineering best practices. I occasionally work with Cypress tests, but work more often with *@testing-library/react*. Database wise, I primarily work with Firestore and write event-driven code that handles asynchronous operations triggered by database actions.
- **Python**: I use a variety of frameworks including Alembic, SQLAlchemy, Django, Django Rest Framework, and FastAPI. I work on features that are less critical in Python, and typically get more time to write and improve our object-oriented code to adhere more closely to SOLID. I work extensively with pytest, but do occasionally work with pytest-bdd and other Cucumber/Gherkin-flavoured tests. Database wise, I mostly work with Postgres. I work with REST APIs that handle synchronous consumer actions, but also work with APIs that interact with Google Pub/Sub and architecture that uses message queues and similar infrastructure.
- **Infrastructure**: I use tools and technologies such as Docker, Docker Compose, and Terraform as well as a variety of internal, bespoke tools.

2021–2024
2y 3m

Head of Engineering
10BE5 · London, UK

I led and managed the engineering team, essentially fulfilling the roles of a lead full-stack engineer, product owner, and pre-CTO. At peak staffing, I had 1 indirect report and 4 direct reports.

Selected achievements

- Produced POC for an algorithmic (without ML or computer vision) method to extract tables from PDFs. This was a highly-requested feature for the Circle-Up product, and was vital for customer adoption. It was implemented in Python, and served via a Flask endpoint. I created full unit tests for this using pytest and third-party PDF libraries, enabling other team members to build this feature using TDD.
- Designed and implemented the majority of the internal logic for numerical data recognition in DOCX and PDF files, part of the core code for the Circle-Up product, and vital for customer adoption. This feature was implemented in Python, and served via a Flask endpoint.

- Using only minimal knowledge of C#, I designed and implemented a C# / JavaScript interop interface to allow a JavaScript web add-in to interface directly with a C# desktop add-in. This feature allowed customers who prefer VSTO add-ins to still use the SendCheck application, while allowing the team to develop one sole application with a thin wrapper. This allowed the team to support multiple deployment targets with reduced effort, improving team velocity.
- Led the development and creation of 3 products from inception (Takedown, SendCheck, Circle-Up), all currently in pilot or commercial phase.
- Shifted development processes from waterfall to Agile. **Reduced average lead time by >75%.**
- Shifted acceptance test ownership to developers, moved team to BDD, and set up standardized code review.
- Empowered developers to propose their own features and take ownership of customer success.
- Trained member of staff with no previous experience in Python to own, design, and implement end-to-end features in Python.

Technologies and Frameworks

- **Product ownership**: I used Notion to records user pains, track customer requests, and plan sprints. I used Agile methodology to ensure that the engineering team met our engineering goals and achieved personal and team growth.
- **Planning + Design**: I used Mermaid to sketch user workflows and draw technical diagrams. I used divergent and convergent design thinking to create feature proposals that met our engineering needs, which in turn were determined by external (customer, user) and internal (business, technical) requirements. I used Cucumber to define acceptance criteria. I used GitLab to break down features into small units of work, using epics to define the overall engineering work required for a user story.
- **JavaScript**: I used Electron, React, and Next.js in JavaScript / TypeScript, using the Node.js runtime. I demonstrated a good working knowledge of TypeScript and the Node.js core libraries in my day-to-day. I guided the team on refactor and React design choices, but deferred to other team members for some specifics regarding best practices in JavaScript and React. I comfortably used CSS to build and edit simple frontend styles, when using the Mozilla documentation as reference. I used Jest to write unit tests for features, and interacted with test code that uses Cypress and Selenium. I used Yarn for package management.
- **Python**: I used Flask and the CPython runtime. I demonstrated a very high command of the language in my day-to-day, and have extensive knowledge of the core libraries. I chiefly wrote algorithmic code, APIs, and functionality for document manipulation (PDF, DOCX). I led the team on code quality and best practices. I used Poetry for package management and deployment.
- **Deployment**: I packaged MSIs for our desktop products (N2N and Takedown) on request. I created Dockerfiles and worked with both Docker and Kubernetes. I owned cloud deployment targets for our other products, which used AWS ElasticBeanstalk and AWS EKS.
- **Infrastructure**: I administrated and own our GitLab repositories, as well as the majority of our Azure and AWS cloud infrastructure. I administrated our AWS infrastructure using both AWS CLI and the AWS console.
- **Quality + Testing**: I used Cucumber to contract our acceptance criteria, and built features that match these acceptance tests. I used Pytest and Jest to build unit tests for public functionality. I followed ATDD, TDD, and BDD, and followed engineering quality principles such as DRY, AHA, and YAGNI. I led the team overall on QA, conducted code review, and led the team on code review standards.

2021–2021
0y 3m

Solution Architect
10BE5 · London, UK

Full-stack development using Python and JavaScript / TypeScript. 1 direct report.

Selected achievements

- Moved engineering team towards GitLab issues for day-to-day work. This required the team to move away from an OSS-style model of merge requests with low traceability towards a model with traceability for changes made. This allowed the team to create better release notes, and improved code understanding and knowledge across the team.

- Created internal packages for packaging mixed Electron + Flask applications to an MSI target. This was a large improvement on the manual packaging done beforehand. Currently used for the Takedown product.
- Created internal packages for quick Flask application creation across products (framework extension).
- Created the backbone for our CI/CD code across all repos, including automated tests and internal documentation (using Sphinx).
- Started to embed unit test culture in the team, moving from low-to-no unit tests towards test as you go.
- Designed more robust git branching strategy (trunk-based development) for new repos.

Technologies and Frameworks

- **JavaScript**: I primarily used Electron, Jest, and the Node.js runtime. I also used React, JQuery, Webpack and Babel.
- **Python**: I primarily used Flask and the CPython runtime. I used pyinstaller and other libraries to create Python binaries.
- **Infrastructure**: I worked a lot with GitLab CI/CD, specifically automated testing.

2020–2021
1y 6m

Software Engineer
10BE5 · London, UK

Part-time role. Primary developer for software packaging/deployment.

Selected achievements

- Created packaging scripts to obfuscate Python code and compile to binaries.
- Created code-signed MSIs for the N2N product.

Technologies and Frameworks

- **JavaScript**: I primarily used Electron and the Node.js runtime. I also used React, JQuery, Webpack and Babel.
- **Python**: I primarily used Flask and the CPython runtime. I used pyinstaller and other libraries to create Python binaries.

2019–2020
0y 6m

Research Assistant
University of Leeds · Leeds, UK

Intensive 6 month project. Automated machine learning using stochastic control methods with time constraints.

Selected achievements

- Created prototype library for automated machine learning.

Technologies and Frameworks

- **Python**: I primarily used scikit-learn, Pandas, numpy, dask, and the CPython runtime. I also lightly used seaborn and Keras.

Education

2015–2020

Plasma Science and Fusion Energy
PhD · University of York
Nonlinear wave-particle resonance in deterministic and stochastic kinetic plasmas
Analytical/computational modelling and analysis of nonlinear systems using linear/nonlinear spectral decomposition; machine learning driven studies of complex data sets in tokamak plasmas; writing high performance parallel code in C.

2011–2015

Physics
MPhys · University of Exeter
Specialisation in QFT, experimental metamaterials, and elasticity theory

Other Projects

derek	Data schema inference	Maintainer
derek-connexion	Connexion (RESTful API) implementation of derek	Maintainer
datatree	Now merged upstream into xarray	Contributor