
CHRIS LAFFRA

Amsterdam, Netherlands • chrislaffra.com • laffra@gmail.com

Developing Innovative Tools for Human Workflows

Work experience

- 2018 - now Staff Software Engineer
Uber, Amsterdam, Netherlands
TLM in the Amsterdam developer platform team.
- 2017 - now Founder
Ahead In The Cloud Computing, Amsterdam, Netherlands
Developing tools for social graph analysis, network visualisation, workflows, and personal information management. Management consulting. Python Data Analytics. Javascript UIs.
- 2014 - 2017 Staff Software Engineer
Google, NYC
Contributed to cloud development tools, internal Google Maps, tools for improving meeting effectiveness, and video conferencing. Self-started projects for discovering relationships between people for performance management, workflow, and work planning using agile scrum board tooling.
- 2010 - 2014 Vice President
Bank of America, NYC
Member of the core team for Quartz, a Python-based strategic platform for real-time risk assessment and reporting across asset classes. I owned the SDLC tooling, platform advocacy, and training. I developed the Quartz Academy, to teach thousands of employees Python and Quartz.
- 2006 - 2010 Architect
IBM Rational, Raleigh, NC
Architected and was chief contributor of EGL Rich UI; a technology to add support for web2.0 and rich user interfaces to IBM's business development language called EGL.
- 2003 - 2006 Senior Software Engineer
IBM OTI, Ottawa, Canada
Optimised the Eclipse platform to run best on IBM's J9 VM, reducing startup time and memory consumption by 35%. Published the Eclipse FAQs book. Improved scalability of WebSphere Application Server by 2X per node.
- 1999 - 2003 Technical Lab Director
IBM OTI, Amsterdam, Netherlands
Founded the lab and led a software development team, working on IBM J9 and Eclipse, leading to the first commercial product based on Eclipse. Represented IBM on JSR075, the Java PDA profile, as part of the Java standardisation process.
- 1997 - 1999 Research Staff Member
IBM TJ Watson Research Center
Performed R&D on Java static analysis tools, leading to tools such as JikesBT and Jax.
- 1994 - 1997 IT Analyst
Morgan Stanley
Developed a UI toolkit in C++ on UNIX and Windows for financial applications called MSTK. Automated conversion of existing C++ code to Java. Published the fourth book on Java. Developed a portfolio management system for private clients.
- 1992 - 1994 Research Staff Member

IBM TJ Watson Research Center

Developed an IDE for the Oberon language and program visualisation tools, such as Hotwire, to visualise the inner workings of Smalltalk and C++ programs.

Education

1988 - 1992

PHD Doctorate Computer Science

Erasmus University, Rotterdam, Netherlands

Designed and implemented PROCOL, a concurrent object-oriented programming language with native support for a proprietary NoSQL database and declarative UIs.

1982 - 1988

M.Sc Computer Science

Vrije Universiteit, Amsterdam, Netherlands

Master's thesis included the development of an expert system to capture the complex set of payment rules across the European union for IBM Finance.

Innovation, Creativity, and Interests

Java Runtime Performance

I am interested in language runtimes and efficient, scalable applications. At IBM, I wrote the Jikes Bytecode Toolkit, to analyse, instrument, and optimise Java applications. Building on that, at OTI Amsterdam, I built a team to work on compression of embedded Java applications, where the entire JVM measures 1MB.

At OTI Ottawa, I wrote memory profilers for the J9 VM and classloaders to make Eclipse start up 2X faster at 1/2 the memory. I also scaled IBM WebSphere Application Server to 2X per node by more effective class file sharing, leading to cost savings in data center hardware.

Learning through Teaching

At Morgan Stanley, I studied Java by authoring the fourth published book on the language. At IBM, to learn about Eclipse, I answered thousands of questions on the Eclipse newsgroups, leading to the Official Eclipse FAQs book. The book was written as an Eclipse plugin.

At Bank of America, I personally educated hundreds of employees in the use of Python and the firm-wide distributed risk, pricing, and reporting framework my team developed. To scale better, I developed an interactive training environment to teach but also test and measure progress. This made adoption of the platform easier saved the firm external training fees.

To prepare for interviews at Google, I developed 50 visualizations of common algorithms and hosted them in an interactive Python training environment on Google AppEngine.

Understanding Workflows

At Google, I developed tools that improve the life of other Googlers. I analysed how people develop software and how they manage information. The result was a graph for all Googlers mapping out relationships between individuals and teams. This information helped improve internal search, internal office maps, and Google's evaluation and promotion process.

As a 20% project at Google, I developed a Chrome browser extension on top of Google's internal bug tracking system. It is used now by thousands of Google teams to run their daily scrum meetings and track their agile development process.

At my current startup, I am integrating lots of information sources, such as email, browser history, chats, etc, to build up a graph of relationships, to help people deal with information overload by organising and prioritising their daily workflow.