415-200-7340 philip.bergen@me.com www.linkedin.com/in/pbergen/

PHILIP BERGEN

Engineer, architect and founder with 30+ years of building tools, systems, high performance teams and companies. My experience covers a wide arrange of business needs: ML, Developer Productivity (with and without AI), CI/CD, SDLC, backend, frontend, big data, embedded, pharma, banking, rockets, and e-commerce. At meta I improved the reliability time wasted by 90%, in the Ads ML training pipeline. That translates to 4-6% improved revenue on a $20B ads algo. At Salesforce I built developer-productivity platforms that grew from 750 to 15,000 engineers, 97% of developers chose my tools, saving 1,000s of FTEs of developer time. Founded Instadia, a web-analytics SaaS several thousand times faster than its competitors, online query in 10+ billion rows on 2004 hardware, was acquired by Omniture. Started out on control systems for Ariane 5 rocket boosters and condition monitoring for Chernobyl nuclear power plant.

Experience (references available upon request)

Meta: Senior Staff Software Engineer (IC7), 2023-2026

ML engineering productivity through automation, tooling, and workflows for the meta Ads algorithm, ($20B/month). I drove the reliability program and reduced Reliability Time Wasted by 90%, shortening the ML delivery cycle by two months. Every month translates to ~3% improved revenue. Tech lead for ML model metrics service and tooling used by about 5,000 engineers, the metrics tooling reduced number of operations by over 60% and saved 20,000 hours per half.

Python, MySQL, Hack (php), sapling (mercurial), VSCode, Linux, buck (bazel), rust, go, claude

salesforce.com: Principal Architect (VP), 2010-2023

Developer productivity scaled from 750 engineers to 15,000. Owned tooling, virtual workstations, VCS, CI/CD. 97% of developers use my tools (voluntary). Helped migrate from ant to maven to bazel. Designed and authored a highly parallel Perforce front end that made it 100x faster while guaranteeing a perfect sync of the 500 GB code base to 7,500 MAU. Setup tools reduced weeks of setup to an hour. Sync tools reduced hours of sync to minutes. Build tools brought build success from 40% to 95%. The savings are in the 1,000s of engineer-years.

My Perforce front-end is open source: https://github.com/salesforce/o4

Python, Bash, Unix, virtual box, docker, openstack, oracle, postgresql, couchdb, Jupyter, pandas, git, perforce (p4), GitHub, GitLab, Jenkins, flask, HyperForce, redis, nginx, nexus, Java, c/c++, perl, maven, ant, bazel, monoliths, microservices, aws, gcp, openstack, macOS, Ubuntu, CentOS

Instadia A/S: Founder R&D, 2000-2007

Big data ahead of most everyone else. Hand-crafted database that handled billions of rows monthly. An amazing SaaS web statistics product, loved by customers and employees. Flexible, capable and performant beyond belief (several thousand times faster than the competition), online query and reporting scouring 10s of billions of rows in seconds. Amazing team of world class engineers.

C++, Java, Python, Ruby, Php, bash, xml, xslt, SunRMI, Corba, Mysql, postgresql, sqlite, javadb, CVS, subversion, Redhat, Debian, SOA

Rovsing A/S: Systems Engineer, 1998

Control systems for the Ariane 5e's rocket boosters (the European Space Agency's biggest rocket). 10x increased throughput in their flagship condition monitoring system for a massive deployment to Chernobyl Nuclear Power Plant.

Matlab, SHARC assembler, Visual C++, Oracle Pro*C/C++, gcc for AD DSP, VxWorks, RCS, Source Safe, Sun OS, Irix, Windows NT

Experience (references available upon request)

Philip Bergen Consulting ApS: **Owner Consultant, 2009-2010**

iPhone development, mostly for a well known Danish pharmacological company.

Xcode, objective-c, photoshop

Wiinz ltd: CTO, 2007-2009

Sports betting arbitrage, developed the backend and control UI for high speed betting arbitrage platform.

Python, Amqp, Postgresql, Perforce, Mac OSX, Redhat

Premitech A/S: Chief Software Architect, 2007

Trust and process. Getting sales, product, support, and development to trust and work together.

Visual C++, MS SqlServer, Source safe, subversion, Windows XP

Adaptive Systems A/S: Systems Engineer, 1999

Built a bank teller system and its printer drivers.

Delphi, Cool:GEN, some inline assembler, Windows NT

Brüel & Kjær CMS A/S: Systems Engineer, 1999

Safety and condition monitoring. CPU board hardware design and driver interface for Windows NT to DSP, AD converters, relays, triggers etc.

Visual Studio C++, x86 assembler, Delphi, Source safe, CMM

Blekinge Institute of Technology (College): Teacher, 1998

Low Level Programming and Microprocessor Architecture.

Motorola 68HC11 assembler, power point

Blekinge Institute of Technology (College): Lab rat, 1997

Built prototypes for my Signal Processing professor's research. The noise cancelling tech ended up in Ericsson handsets.

Embedded (no OS, just assembler), TI DSP assembler, soldering iron, lithography for producing circuit boards

Cykelbergen AB: Owner, founder, mechanic, 1993-1997

Probably the world's first bicycle e-commerce.

C, CGI, Unix, Allen keys

Education

Blekinge Institute of Technology (www.bth.se/eng) - Electrical Engineer 1993-2006

Dropped out for a job with Rovsing doing actual rocket science. Later came back and finished my bachelor's degree, but alas due to a clerical error all I can claim is a University Diploma with specialization in Electrical Engineering (120p).

Other Skills

Fluent in English, Danish, and Swedish, conversational French.

Patents

SYSTEM AND METHOD FOR DETECTING AN ERROR IN SOFTWARE

US 10,423,522 · Issued Sep 24, 2019

A computer-implemented method of detecting a likely software malfunction is provided. The method comprises collecting a plurality of software error data sets wherein each software error data set comprises a proposed code section containing an error and a corrected code section containing code changes that fixed the error in the proposed code section. The method further comprises training a computer-implemented algorithmic model using the collected software error data sets to devise a software code classifier for predicting a likely error in a code section, reviewing a section of code using the software code classifier, and identifying suspicious code in the reviewed section of code as containing a suspected error using the software code classifier.

Direct Build Assistance

US 10,114,627 · Issued Oct 30, 2018

A tool for installing and configuring a software development environment with an embedded help feature is provided. The help feature may provide a user (e.g., developer) with technical assistance to resolve problems that occur when building software. In one example, the system may include a configuration tool that includes a help feature and installs and configures one or more software programs to build software. The software programs may include, for example, operating systems, source code editors, debuggers, software build tools or any component of a software development environment.