Trevor C. Kemp

Pittsburgh, Pa. http://trevorkemp.work trevor@trevorkemp.work

Software Development Professional

Skilled in effective communication and adept at leading and collaborating with diverse teams across various geographical locations. Proven ability to quickly grasp new concepts and technologies, demonstrating a strong commitment to continuous learning. Known for creative problem-solving abilities and flexible approach to challenges. Experienced in coordinating with partnering engineering firms and clients, serving as a technical liaison between technical and non-technical stakeholders. Proficient in a wide range of technologies, spanning from embedded systems to user-facing products.

Skills

People Management - Software Architecture - Software System Design - Software Performance Optimization - Software Project Management - Software Development - Software Testing - Real Time Systems - C++ - C - Linux - Linux Kernel - Python - Bazel - AUTOSAR Classic - Java - TCP/IP - TLS/SSL - Basic Zero Trust Security Architecture

Professional Experience

- *Latitude AI*, 11/2022 - Present (acquisition of Argo US by Ford Motor Company)
  Staff Software Engineer
  Senior Software Engineer
  Presently serving Latitude on the Autonomy Integration team, producing Ford's next ADAS.
  - Served as HIL bench tech lead for 12 months
    - Visionary and lead of a cross-team group of 20+ people through a dotted-line reporting relationship to 2 managers.
    - Took Ex-Argo/Latitude from 1-2 manual benches in constant disarray to a pool of 18 automated + 3 manual benches through introducing basic configuration management, standing up triage processes, and automating the reporting categorization of test failure causes.
    - Primary developer of 3 company-wide testing workflows and related tooling. Through my work, we have reduced the need for 100% of integration testing on vehicles to nearly all integration testing occurring on benches. Reduction from several hours to days of feedback time at multiple $100s/hour to ~30 minutes and less than $4/test.
  - Serving as integration architect, reviewing designs across multiple teams, and continuing to design custom development testing workflows.
- *Argo AI*, 05/2020 - 10/2022 (close of company)
  Senior Embedded Firmware Engineer, 01/2022 - 10/2022
  Software Engineering Manager 05/2020 - 12/2021
  Served Argo in an embedded developer capacity on the Argo Lidar sensor development team
  - Lead developer, application architecture & development. Produced and executed plans for the team to go from 0 to functional in 4 months, down from 12 months under the prior plan. Emphasized incremental improvement and de-risk over run-to-completion, to always have a working product.
  - POC for application architecture and safety review.
  - Bar-raiser: mentored other engineers (junior & senior), teaching healthy design, coding, and testability patterns against Argo coding standards and effective use of modern C++. Emphasized design for testability.

  Served Argo in a management capacity on the Autonomy Runtime Team for Self Driving Vehicles.
  - Simultaneously managed two teams of 5 people each across 3 hours of timezones, supporting

other teams across 9 hours of timezones: onboard heavy data logger, onboard autonomy system health monitoring framework.
    - Technical roadmap development, cross-team coordination, weekly reporting to VP.
    - Hiring and team development, personnel development, coaching developers through difficult situations, performance reviews & growth plans. Oversaw promotions of 2 engineers.
    - Multiple high-quality or key technical features
        - Drove dramatic reductions in time required to offload heavy data logs, going from 40 minutes to fitting in operator break periods.
        - Reduced on-road data loss events from 2 per quarter to none per quarter, requiring work with multiple teams, including non-engineering teams.
        - Established verification strategy for deploying log reader changes, reducing major regressions from 2 per quarter to none per quarter, and demonstrated a quantified need for offboard pipeline owners to establish automated tests.
        - Supervised 12GB reduction in onboard runtime memory usage in the onboard infrastructure.
        - Oversaw the implementation of a key ftrace-based concurrency performance debugging tool.
        - Consistently lauded by peer managers and leads for bringing strong organization and effective responses to internal customer requests, better than any prior lead of these efforts.

- **Senior software Engineer**, *Uber Advanced Technologies Group*, 10/2018 - 05/2020
  Served Uber in an engineering capacity on the Motion Planning team for Self Driving Vehicles.
    - Designed & developed an automated configuration management mechanism in Bazel, C++ 14, and Python to provide technical requirements traceability across the autonomy stack and verification systems, bolstering safety and reducing opportunity for simple failures that killed releases.
    - Member of a small team to optimize the Motion Planner's onboard performance and software architecture. Reduced planner latency by 18% at 50% tails and 15% at 99% tails by removing superfluous predictions from consideration.

- **Principal Software Engineer (Architectural & Lead Capacities)**, *Sophos,* 04/2017 - 10/2018.

Served Netronome and Sophos during a team acquisition in a technical lead capacity of 6+ engineers on a next generation firewall.

    - Architectural design to bridge kernel, offload device, and user space for several key features.
    - Linux network stack & NUMA-aware kernel module + userspace library implementations.
    - Set & maintained high expectations for developers: introduced code review, design writing & review, sound testing methods.
    - Planning & intracompany cross-team roadmap development; mitigated problems by altering roadmap plans, and demonstrated how slip would affect the overall product's schedule.
    - Modified scrum-style sprint planning, cross-team coordination, and progress tracking. Introduced sprint planning & grooming sessions for planning upcoming work to dramatically reduce entropy and churn. My team's estimates were the most accurate in the business unit.
    - Personnel development: chose efforts for engineers to learn new skills, expanded their engineering abilities, and cross-trained members to reduce project risk and permit more agile team structures.

- **Software Engineer/Senior Software Engineer**, *Uber Advanced Technologies Group,* 08/2015 - 04/2017.

Served Uber on a team developing products in mapping, safety, and autonomy systems.

    - Primary owner of Linux operating system on self-driving cars. Supported multiple kernels for matching hardware configurations. The OS build framework was in Python.
    - Automatic compute node provisioning for manufacturing 1,000 self-driving cars, in Python. Reduced installation time by 15 minutes and removed all opportunity for costly mistakes by technicians. The board type and revision were detected and the appropriate kernel, drivers, and node identity were installed.
    - Diagnosed multiple system-level SDV issues (lost network packets under load, identifying and fixing complex latency problems, limiting effects of memory leaks). RT Linux system tuning for latency

  reduction.
  - Improved technical communication by introducing a design document template that was adopted org-wide for all software design.
  - Wrote and integrated several sensor interfaces in C++11 and debug utilities for self-driving cars.

- **Software Engineer**, *Netronome Systems*, 06/2013 - 08/2015.

Served Netronome and partnering firms by assisting and leading in the design, implementation, testing, and maintenance of network flow processing systems:

  - Assumed project lead responsibilities on TLS/SSL inspection library, written for Linux in C and C++. TLS session termination/MiM for tens of thousands concurrent SSL flows. TCP/IP (protocol-level), TLS/SSL core protocol and necessary extension. Primary PoC for client and Netronome management for project.
  - Trusted developer on a key product. Gathered requirements, key assistant in features' firmware design, implemented and integrated TCP checksum offload and Large Segment Offload on Netronome NFP 6000 inside VXLAN and NVGRE overlays.
  - Assistant programmer/researcher on RDMA implementation. Implemented Infiniband building blocks on Netronome NFP 6000 with a team of industry experts. Learned basics of Linux PCIe driver implementation.
  - Decreased RDMA project entropy by working with project lead to establish test-driven-development and continuous integration practices.
  - Trusted communicator and software developer; presented multiple project status and roadmap updates to Netronome and customer executives.

- **Software Engineer**, *Compass Systems, Inc***.**, 05/2009 – 06/2013.

Served on multidisciplinary teams by assisting in the design, implementation, testing, and maintenance of applications and systems including:

  - System, application, client/server applications, device integration, Linux kernel compilation and configuration, and Linux kernel GPIO driver software development in ARM + Linux, C, C++, GNU Make, Qt, SQLite, BASH, for a handheld system for mobile geo-data collection.
  - Application employing Windows, C# .NET, MySQL, Python, and a custom XML schema to collect, store, and transfer geo-data to a client system and produce custom reports.
  - Implemented in C++, implementation of the Tactical Automated Security System standard. Message routing for a heterogeneous network of microcontrollers (i8051, ARM + Linux, Rabbit 3000).
  - PoC for clients and partners on several software-technical projects, including Compass Systems' appearance at the 2011 Coalition Warrior Interoperability Demonstration.

Education

- **MS, CS,** West Virginia University, Graduated May, 2009.
- **BS, CS. Applied Mathematics Minor**, University of Pittsburgh, Johnstown, Graduated May, 2007.