

# Russell Brooks

**Phone:** 267-500-4300  
**LinkedIn:** <https://www.linkedin.com/in/russell-brooks-b46b4414>  
**Email:** [me@russbrooks.com](mailto:me@russbrooks.com)  
**www:** [www.russbrooks.com](http://www.russbrooks.com)  
**GitHub:** <https://github.com/iq9>  
**Blogs at:** <https://dev.to/iq9>

## Objective

Work with exceptional people to delight the customer with useful software that solves clear problems.

## Experience

<b>Senior Engineer, Data Engineer</b> <a href="#">Trainual</a> • Scottsdale, AZ	2020 - 2023
<b>Senior Software Engineer</b> <a href="#">ShopRunner</a> • Conshohocken, PA	2017 - 2020
<b>Sr Eng, Dir of Software Engineering</b> <a href="#">TelVue</a> • Mt. Laurel, NJ	2009 - 2017
<b>Senior Software Developer</b> <a href="#">Safari Montage</a> • Wynnewood, PA	2005 - 2009
<b>Senior Software Engineer, Design Team Manager.</b> NetReach, Inc. • Ambler, PA	1999 - 2005
<b>President</b> Atlantic Image, Inc. • Wildwood Crest, NJ	1995 - 1999

## Skills

### Software Craftsmanship

- **Server-Side Programming** - 30 years total. 20 years on Open Source: Ruby, Python, ES6 (Node), TypeScript (Deno), Go, Rust, and frameworks such as Fresh, Rails, Sinatra, Express.js, and Flask.
- **Client-Side Programming** - High-level expertise in ES6 and TypeScript on React.js. Asynchronous programming techniques, at the View level, to amplify UI responsiveness and user experience.
- **Patterns, Architecture, and Design** - Fluent in multi-tiered architectures such as “Model, View, Controller” (MVC). Devout practitioner of DRY, SRP, SOLID, and OOP principals to maximize code reuse and minimize coupling between

components, increasing code maintainability and lowering cost of development. In continuous study of the proven Patterns of software design.

- **Testing** - Test/Behavior-Driven Development (TDD/BDD) with libraries like RSpec, MiniTest, Jest, and native Rails Unit, Functional, and Integration. Test Coverage when I arrived at TelVue: 10%. Test Coverage now as a result of my initiatives: 90%
- **Asynchronous Background Processing** - Advanced knowledge of server-side background-job processing, using technologies like Sidekiq (Ruby), RQ (Python), queueing to Redis.
- **Project Management** - Proponent of Agile software-development methodologies and iterative development. Replaced Waterfall process with “Scrum” at TelVue, resulting in a 25% dev efficiency increase. Rolled out new initiatives such as Pair Programming, Coding Standards, and Peer Code Reviews, which translated to a direct increase in software quality.
- **Documentation** - Excellent communicator, technical writer, and meticulous software-documentation author. (GitHub Discussions, Confluence, MediaWiki)
- **Service-Oriented Architecture (SOA) and MicroServices** - Adept at integrating disparate systems securely and reliably across insecure, unreliable, public networks - for example, JSON REST Web Services over HTTP(S).
- **Security** - Deep knowledge of latest security practices and hardening systems against attacks: TLS, cryptography, and integrating user-bases across systems: Auth0, SSO, GCP IAM, Workload-Identity Federation (WIF), etc.
- **Software-Configuration and Lifecycle Management (SCM & ALM)** - Thoroughly versed all aspects of revision control (Git), branching models, codeline policies, continuous integration CI/CD, defect tracking (Pivotal, Jira), and release management.
- **Database Administration and Programming** - Primary technological focus for last 20 years: PostgreSQL. 8 years experience writing and optimizing raw SQL queries, stored procedures, and UDFs in other SQL DB's such as MySQL, SQLite, and MS SQL, GCP Cloud SQL, AWS DynamoDB, Aurora, RDS, and Redshift, as well as performance tuning and data integration between different systems. In depth knowledge of Memcached and Redis.
- **Durable Queues / Message-Oriented Middleware** - 6 years experience with RabbitMQ (AMQP).
- **Serverless**: 5 years experience with Serverless products such as AWS Lambda's and GCP Cloud Functions.

## Russell Brooks

---

- **UNIX Sockets and Shell Scripting** - Advanced knowledge of UNIX & Linux CLI tools (Shell Script, SED, AWK, etc), process management, and interprocess communication.
- **Content-Management Systems (CMS)** - Rolled out ~10 instances of Drupal, WordPress, and MediaWiki, then maintained them.
- **Open-Source Contributor**
  - [SQLFluff GitHub Action](#) - Added a feature that commits directly.
  - Rvideo gem, [Nesta](#) static-blogging system, Homebrew, StreamIO FFmpeg.
- **Open-Source Software Author**
  - “[Alact](#)” Ruby gem. Lossless-music conversion utility for audiophiles. 32,693 downloads.

### Digital Video

- 12 years professional experience with Digital Video, IPTV, and Web Streaming. 9 years experience specific to Television Cable industry.

### Management

- 15 years experience managing Enterprise-scale Web projects of varying disciplines and budgets, with strengths in:
- Assembling and managing multiple project teams, and teams of consultants, and motivating them toward project goals and deadlines.
- Crafting strategic direction of large-scale Internet and Intranet projects, based on market analysis.
- Iterative stakeholder communication to manage expectations and change orders, report on project milestones, educate, and gather and implement requirements.
- Forecasting and managing large project budgets.
- Collaborating with QA personnel, formulating and documenting test plans and strategies for functional, integration, regression, and unit testing.

### Design and UX

- 25 years experience creating feature-rich, usable, Web applications and graphical user interfaces using Photoshop (Adobe Certified Expert), and hand-authoring markup to standards. Very well-versed in user-interface design, web usability, layout, color theory, typography, iconography, and design principals. I am in a relentless pursuit of simplicity.

### Systems Administration, DevOps, and IAC (Infrastructure as Code)

## Russell Brooks

---

- Advanced knowledge in the configuration, maintenance, virtualization, monitoring and scaling of \*nix platforms running on Nginx, Puma, and on Cloud services such as AWS and GCP.
- 7 years experience with Container-based Architecture managed by Docker, Docker Compose, Kubernetes, and GKE.
- Guru-level skills in IAC using Terraform, Chef, Ansible.

### Data Engineering

- 2.5 years experience with working on a Business Intelligence team. (BigQuery on GCP).
- ETL
- [dbt](#) SME at last employer. I brought that tech to the company.

### Projects

#### Business

**Trainual** – <https://trainual.com/> - 50,000 active users

**Period:** 2020 - 2023

**Project Budget:** \$50m

**Technology:** Ruby 3.1 on Rails 6, but Python in the Data-Sci dept I was in.

- **On DevOps team that moved their cluster from Heroku to GCP, and scaled it.**
- **On Data Eng team that set up and rolled out their ETL Pipeline.** GCP BigQuery, dbt, FiveTran, Airflow, Heap, Terraform, Semaphore, Segment.io.
- **Scaling**
  - Scaled the Kubernetes (GKE) cluster: Added auto-scaling to Nodes that needed it, and researched and recommended faster / cheaper things.
- **Serverless**
  - Wrote a GCP Cloud Function in Python - It shuttled some data around on a schedule.
- **[dbt](#) Rollout and Reorg**
  - Lead the charge with adopting this. I immediately saw immense cost-control value in this tech, having lived thru the horrors of SQL Stored Procedures.
  - When I inherited the dbt project, there was **1 Model** and no Conventions being adhered to. The Engineer in charge of it left. Thanks to my work, 1 year later,

## Russell Brooks

---

there were over **500 Models**, and a radically improved directory, file, and [DAG](#) structure, that adhered 99% to dbt's official standards.

- Brought 3 teams (Eng, Data Sci, and BI) up to speed in its use, (in code and on dbt Cloud), good patterns, and documented it all meticulously to the wiki.
- Researched and implemented ways to improve performance of dbt jobs.
- Researched ways to protect the parts of the repo that break Analytics when the Model names are changed. Found that dbt "Exposures" aims to solve that. Educated the team.
- Wrote the company's first dbt Macro, and educated the teams on how they work and left an example.

### • **SQL Linting**

- Researched and evaluated the major SQL Linters. Arrived at [SqlFluff](#) as the winner, and got buy-in from the teams.
- Worked with the 3 teams to converge on exactly which Formatters, Conventions, and Dialects to set it to.
- **Rolled it out on CI** (wrote a GitHub Action) in the dbt repo. It would make GitHub "Suggestions" on PRs. A nice cost-control touch. Contributors could simply click "Accept" if they agreed with the change.
- Per the team's suggestion, added some functionality to this 3rd-party GitHub Action, and submitted a [PR for that](#). It commits the changes directly, saving PR authors more time. Trainual's **first Open-Source contribution**.
- Researched and eval'ed IDE extensions for SQL Fluff. Recommended the best ones to the team. This allowed 1-click formatting convenience, cutting more costs.
- **Secured several things with a bulletproof, modern, Auth mechanism called "[Workload Identity Federation](#)".**
  - SQL Fluff, when set to "dbt" Dialect, has to make a REST call to BQ to read some metadata. It needs to do this on CI and locally on Dev machines. I added "Workload Identity Federation" (WIF) to those calls on CI (a GitHub Action). WIF had never been done before at Trainual.
- **Realtime User Monitoring (RUM) - DataDog (DD)**
  - Segment errors were not detailed enough. They were going into a single bucket thanks to a generic Error msg. I discovered that the Segment gem

## Russell Brooks

---

provides 4 more values on errors that we weren't tracking. Learned the DD gem and added these as custom Attributes to the errors.

- **Customer data platform (CDP)** - [Segment.io](https://segment.io)
  - Ramped up on this tech quickly, he had never used before.
  - Added about 50 new Analytics calls from the main Rails application, Sidekiq jobs, and the React.js parts
- **Main SAAS app (Rails) and its Admin site**
  - Upgraded the main Application to Ruby 3.1. Brought the department to speed on the next-gen debugger that is bundled with Ruby 3.1. This allowed us to eliminate 2 dependencies from our Gem stack: Pry and ByeBug. The department was very pleased with that DX (Developer eXperience) improvement.
  - Added a number of new features, including their innovative "Business Playbook". The frontend is TypeScript on React.js.
- **Plugged several Security vulnerabilities in front and backends.**
- **Cut Dev costs by 30% by pentupling transpile times with Vite.**

**ShopRunner** – <https://www.shopperunner.com/>

**Period:** 2015 - 2019

**Project Budget:** \$100M

**Technology:** Python on Flask, AWS, Postgres, Kotlin, Node, TypeScript, React

- **Serverless** - Wrote **AWS Lambda** in **Python** with the **SqlAlchemy ORM**. It was a Microservice that returned the Calendar holidays. This eliminated a lot of duplication and inconsistency. There were 6 different Holidays implementations prior to that. Now there was 1 shared across all the services.
  - I updated those 6 systems to utilize it. A mix of client-side JS, TypeScript on React, Kotlin, Go, and Python.
- **Added features** - applications, both internal and external. Vanilla JS, and TS on React connected to Python/Flask microservices.
- **Debugging** those same applications. Prod issues.
- **Security** - There were some **public** microservices that had **no Authentication**.
  - Enlightened the department to [JWT tokens](https://jwt.io). I hosted a LnL on how they work.

## Russell Brooks

---

- A teammate and I wrote an Auth-Token Server that sat outside the Firewall to grant JWT tokens, and expire them after 1 use.
- We added that security to the Public endpoints (Python on Flash).
- **Brought Deterministic Builds to the company** with [PipEnv](#).
  - Improved Developer experience by making dependency management simpler in Python. No more wasting dev costs manually managing Pip's Virtual Environments. It now was abstracted away from us.
- **Researched and stood up their first Background Job server.** I saw a clear need for it immediately. They had shoehorned RabbitMQ into background-job duty. RMQ has no progress-reporting features, so you can't show the end user a progress bar of job progress. It's a message broker; not a background-job server.
  - **Discovered [RQ](#)**, a lightweight Python one similar to Sidekiq. Queued to Redis.
    - **Did a LnL.** Got buyin from dept.
    - **IAC** - Added it to the all clusters by way of [Ansible](#) / Terraform.
- **Scaling with Kubernetes / Docker.**

**TelVue Connect** – [www.telvue.com/products/telvue-connect/](http://www.telvue.com/products/telvue-connect/)

**Launch:** June, 2012

**Project Budget:** \$2.0 Million

**Technology:** Ruby 2.3 on Rails 3.2, Python on Flask

- Cuts a Cable company's "content-ingest" costs in half by streamlining a highly disorganized and costly workflow in their business.
- Architected the entire stack myself. It was a radical improvement on what was there at every layer. (MySQL 👉 **Postgres** | RedHat 👉 **Ubuntu** | Delayed Job 👉 **Resque** | Jobs queued in the RDBMS 👉 **Redis** | Apache 👉 **Nginx**)
- 3 months coding and designing the MVP myself before assembling the project team.
- All UI Design by me.
- Wrote (from scratch using no libraries) a very innovative UI component in vanilla JavaScript - a DnD, multi-file uploader, with progress bars, very-large-file support, and resumable-upload capabilities. That combination of features in an uploader had never been done before, in a browser-native way, anywhere on the Internet.
- Transocode Farm - Wrote all background-job pieces on Resque and Redis.

## Russell Brooks

---

- As Project Manager on this project, I gathered and documented all requirements, from stakeholders, coordinated 20 developers across 3 consulting firms and internal employees, delivering 1.0 on time and under budget in under 1 year, using Agile processes.
- 90% test coverage
- 100% Uptime for a period of 9 months as a direct result of my work. This was the first time in the company's history this level of uptime was achieved.
- Awarded "[Best Innovation in Content Management Systems](#)" by the TelcoTV Vision Awards (largest conference in this industry).
- Awarded [4 out of 5 stars](#) by Broadcast Technology Report.
- Finalist in Communication Technology's 2012 Platinum Awards.
- Product has been very well received. Over 5,000 organizations have signed up since launch. Comcast is operating a portion of their infrastructure on TelVue Connect.
- Scaling and Performance - Lead the scaling initiatives by breaking up the application into more dedicated components. Implemented a Redis Caching Server and wrote the Rails Fragment Caching pieces resulting in significant performance increase. Implemented Streaming Replication on the Postgres 10 Database cluster. Performed a complete SQL performance analysis. Now no page, application wide, returns in over 2.5 seconds, whereas some were taking over 60 seconds prior.

Added functionality to several other products in TelVue's suite, including [HyperCaster](#) and [CloudCast](#).

**SAFARI Montage** – [www.safarimontage.com](http://www.safarimontage.com)

**Launch:** August, 2005

**Project Budget:** \$2.5 Million

**Technology:** PHP, Postgres, FreeBSD and Linux

- SAFARI Montage is a groundbreaking, educational-video server for schools nationwide. It delivers high-quality DV to classrooms via a sophisticated browser interface. I was hired as a lead Software Architect and Project Manager to design, build, and drive this project through to completion. We delivered version 1.0 on time. There are currently ~40,000 units in place throughout the U.S. and Canada within the 7 years since launch. The content includes recognizable producers like National Geographic, Discovery, The History Channel, Disney, Reading Rainbow, and PBS. The system has won several prestigious industry awards and was rated the "top product in its class" by the School Library Journal. Actor, Richard Dreyfus, has been a prime spokesperson for this product.



## Russell Brooks

---

- Listed as “Inventor” on International [Patent # 2008/005874](#) and European [Patent # EP2054798](#), for innovative components I co-engineered for this product.
- Architected and helped code the distributed version for WAN environments. Hundreds of Montage servers can be connected over traditional Internet connections, sharing videos, playlists, and other personal data throughout an entire school district.
- Architected, designed, and programmed numerous features within the system such as search and advanced search, digital-asset management, drag-and-drop playlist foldering system, downloadable content, permalinks feature, and dynamic user-interface enhancements using the jQuery JavaScript libraries.
- Performance optimization – SQL queries, code, and browser-side performance

### **SAFARI Montage HD Network** - [www.safarimontagehd.net](http://www.safarimontagehd.net)

**Launch:** June, 2009

**Project Budget:** \$500,000

- An online community website, architected and built almost entirely by me within a **10-week timeframe**. I lead a four-person project team, but the other three resources amounted to ~10% utilization over the project life-cycle. They were used primarily during two-week sprint prior to launch to ensure deadline was met.
- Launched precisely on deadline, in time for unveiling and launch banquet with partners and stakeholders at a key industry tradeshow.
- I took on all Sys Admin duties, including Development, Staging, and Production environment setups, security configurations, and PostgreSQL database server setup. The site is PHP/Drupal-based so it is running in Apache on Ubuntu Linux.
- Through the use of Blogs, Forums, Lesson-Plan sharing features, the HD Net site closed a critical feedback loop between LVC, its partners and customers. It provided LVC with a real-time view into its customers’ needs, so it could support its customers more effectively and adapt rapidly to market conditions – equating to an estimated \$500k/yr profit.

### **Philadelphia Eagles** - [www.philadelphiaeagles.com](http://www.philadelphiaeagles.com)

**Launch:** April, 2001

**Annual Budget:** \$320,000

- Point person for all client relations with key contacts at the Philadelphia Eagles organization.
- Led an 8-person team of developers and designers to redesign, rebuild and maintain site for the 2001 thru 2003 NFL seasons

## Russell Brooks

---

- Delivered a completely rebuilt web site within seven-week deadline and remained within budget
- Architected a browser-based content management system that allowed the client to manage 95% of their site content, during live NFL games, from stadiums nationwide
- Built a successful online store for merchandise and collectible sales generating \$1.4 million in annual revenue
- Developed Banner Ad and Contest Management applications, generating \$400K - \$600K in annual revenue

**Widener University** - [www.widener.edu](http://www.widener.edu)

**Launch:** August, 2003

**Project Budget:** \$650,000

- Successfully managed the architecture and design phases of the project, remaining strictly within budget.
- Conducted formal presentations of architecture and design compositions for Deans and Director-level contacts in lecture halls at the university.
- Architected and developed a new CMS used by all schools University-wide, that moved authoring control to non-technical personnel, cutting website update labor & time in half, saving the school significant costs.
- Managed the reconstruction and migration of over 5,000 pages of existing site content

### Personal

**Personal project to learn the next-gen tech, Deno on Fresh.** <https://www.russbrooks.com/> My personal site, rewritten and redeployed on May 2023.

**Personal project to learn “Nest.js” Node framework and TypeORM** - [Backend source code](#). The Component-based [Frontend source code](#).

**Personal project to bring my Node “Express.js” framework skills to date** - [Backend source code](#). The Component-based [Frontend source code](#).

**Personal project to bring my Ruby and Rails skills to date** - [Backend source code](#). The Component-based [Frontend source code](#).

- Ruby 2 on Rails 5 MicroService backed by SQLite.

**Personal project to learn Rust and Rocket.** [Backend source code](#).

- Rust 1.43 on Rocket 0.4.2 MicroService backed by Postgres.

### **Portfolio**

Please see my personal site for a sampling of my work over the past 30 years. This is only a fraction: <https://www.russbrooks.com/>

Many sites in my portfolio were created entirely by me, including the design, database, and programming. Nearly all of the newer sites have hand-crafted Content-Management Systems. I have created or contributed to well over 60 Web applications since the Web was born, from corporate presences, to medical portals, to enterprise-grade Intranet solutions.