

Douglas Fisher

Software Technology Development & Leadership

This document is an export of <https://dfisher95350.github.io/>.

Please see that site for the very latest and additional material.

Summary

- Professional software leader, manager, and developer of many years experience.
- Leadership roles in six Independent Software Vendors (ISVs), including two greenfield startups.
- Shepherded a greenfield ISV startup to acquisition and subsequent growth.
- Diverse experience as a technical consultant for three consulting groups.

Contact

Douglas Fisher

douglas.jon.fisher@gmail.com

612-325-3826

GitHub: <https://github.com/dfisher95350>

LinkedIn: <https://www.linkedin.com/in/douglas-fisher-0a43971>

Adobe Creative Cloud: <https://www.behance.net/dfisher95350>

D.B.A. Douglas J Fisher LLC, MN, USA.

Resume

The shorter version of my resume. (TLDR). My skills, experience, employment, and education in a nutshell.

Skills

Abilities where positive feedback says I've been able to make exceptional contributions.

- Bridging the gap. Software product development is the discipline of translating product needs into technical implementation. Reducing friction at this boundary is a distinct competitive and innovative advantage. And yet product and technical disciplines are traditionally silo'd. Most people don't even want to have expertise in both. I understand and appreciate both disciplines.
- Depth and breadth. Able to adapt communications and contributions appropriate to the current audience or situation, from hands-on pair programming to high-level board presentations. Useful in the weeds. Useful at ten-thousand feet.
- Keeping it fresh. Don't plan to solutions but rather to solution vectors, so that technology choices can always be evolving in a forward direction.
- Keeping it real. Use pragmatic, proven technical solutions for real-world problems. Make smart build v buy decisions.
- Making it personal. Customers matter. Encourage direct interaction and make it as enjoyable as possible for everyone, particularly for developers who don't always love this aspect of the job. Nurture a sense of responsibility and respect.
- Mentoring. Quality developers don't need hand-holding, but they do need to be set up for success. They need help setting meaningful goals and expectations. They need help identifying the most productive tools (and patterns) for their toolboxes.

Experience

Management

All the things that go into managing product development at an [Independent Software Vendor \(ISV\)](#).

- Agile. Although the [Agile](#) Manifesto was first published in 2001, it was an expression of the development climate of the late 90s. As such, the Agile methodology has been standard operating procedure throughout my career.
- Compliance. Experience with high-level procedural audits, such as [ISO 27001](#), [SOC](#). Experience with implementation-level standards, such as [OWASP](#), [OAuth](#), [SAML](#).
- Continuous Deployment. More than a decade of authoring and managing deployments in the Amazon Web Services (AWS) ecosystem.
- Continuous Integration. A long history of designing, implementing, and managing technical solutions and procedures around various popular continuous integration systems, most recently GitHub Actions.
- Planning. Very familiar with Agile planning tools and techniques. Experience in Enterprise Architecture provides an additional layer of experience, in frameworks such as [BIZBOK](#), [COBIT](#), [ITIL](#), [ITSM](#), and [PMBOK](#).
- Product Management. Experience implementing the [Pragmatic Marketing](#) product development framework.

- Project Management. A long history of documenting, prioritizing, and managing, with tools such as Atlassian suite (Jira), etc.
- Quality Assurance (QA). A long history of designing, implementing, and managing technical solutions and procedures around various popular unit testing and Quality Assurance (QA) frameworks, e.g. JUnit, Jest, Selenium, TestRail.
- Version Control. A long history of designing and managing solutions and procedures around various popular version control and artifact management systems, including of course Git and GitHub.

Languages

Hands-on object-oriented, aspect-oriented, and functional programming (and the design of complex systems that leverage each), from the late 90s and continuing through the present.

- Java, since version 1. Name any major language or framework that runs on the Java Virtual Machine, and I've probably worked professionally with it. For example, Kotlin and Spring.
- JavaScript, since before it was called JavaScript. I've worked with most of the frontend frameworks, and some of the backend. I work with TypeScript and React, for example.
- To a lesser extent, I've also worked professionally in the .NET (C#), Python, and Ruby ecosystems.
- I dabble in C++, Go, and Rust. Way back I did Perl professionally.

Data

The catch-phrase, "data-driven", has never been more relevant as the present. A staple of my career, and the aspect of software product development I care most about.

- SQL-compliant databases, since 1998. MySQL, PostgreSQL, Oracle, Microsoft SQL Server, IBM Db2, IBM Informix, etc.
- [Document stores](#) (particularly for [large-scale search](#) solutions), since 2000. Apache Lucene, Apache Solr, Elasticsearch.
- [Graph databases](#) (Neo4j, Cypher) since 2015.
- Object-relational (ORM) frameworks (e.g. Hibernate, Spring Data), since 2002.
- Business Intelligence tools and frameworks (e.g. Tableau).
- Data visualization (e.g. D3).
- I dabble in data science (e.g. NumPy, Pandas, Anaconda) and machine-learning (e.g. TensorFlow).

Architecture

Advanced to "technical architect" early in my career, and have been designing systems ever since.

- Software Design. Frameworks, applications, platforms. Using [UML](#), [ERD](#), [BPM](#), etc. since 1998. Most often with hands-on proofs-of-concept.
- Enterprise Architecture. Experience as both a consultant and the manager of an EA SaaS product. Familiar with TOGAF and similar guiding frameworks.

- Virtual Infrastructure. Hands-on experience designing and implementing solutions in Amazon Web Services (AWS), Docker, Kubernetes, Vagrant. VMware before that.
- Physical Infrastructure. Even today comes in handy when understanding cloud-based [hypervisors](#) and threading models.

Operating Systems

- Linux, of course. I've worked with most major distributions since the mid-to-late 90s. POSIX-compliant macOS since 2010. [Windows Subsystem for Linux](#) since 2020.
- In the way-back I worked with most major flavors of Unix (including hardware infrastructure planning).

Cloud

- Amazon Web Services (AWS). Since 2010 this has been my main professional deployment environment. Of course this means that I've worked with many of their sub-systems and services.
- I've dabbled in Azure, Google Cloud Platform (GCP), Heroku, and others.

Employment

In a nutshell, I've been a technical leader (architect, chief technologist, manager) since I really found my feet at SPS Commerce, circa 2000. I've been building Web applications since 1995, Software-as-a-Service (SaaS) since 1998, and cloud-based (e.g. Amazon Web Services) systems since 2010.

- My own software incubator, since 2021. Includes a major [fintech](#) integration project ("Civis Mundi") to make personal banking history available across borders so that immigrants need not get stuck into the exploitative high-interest credit card game until they can develop domestic credit history.
- VP Engineering, Planview/Changepoint, 2015-2021. Independent Software Vendor. After the acquisition of barometerIT, helped Changepoint, and then Planview, evolve and grow the platform.
- CTO and Product Manager, barometerIT, 2011-2015. Greenfield startup. Developed a SaaS product for Enterprise Architecture management. Key customers included a number of Fortune 100 companies. Guided the company to acquisition via engagement with key analysts (e.g. Gartner).
- Chief Technologist, International Decision Systems, 2008-2011. Independent Software Vendor. Contributed to significant decisions regarding the transition to next-gen software development patterns and platforms. Introduced a new technical foundation still in use to this day (last checked 2022).
- Senior Technical Architect, Perficient, 2006-2008. Consultancy. Various technical consulting projects, particularly with respect to WebLogic and WebSphere Java application platforms (J2EE).
- Technical Architect, MSI Systems Integrators, 2005-2006. Consultancy. Various technical consulting projects.
- Chief Architect, Notiva, 2002-2005. Greenfield startup. Developed a SaaS product for B2B supply chain transaction reconciliation. Key customers included Target Corp.
- Technical Architect, SPS Commerce, 1998-2001. Independent Software Vendor. Developed a post-EDI platform to enable small and medium-sized businesses to participate in large-scale transactional B2B supply chain integrations.
- Technical Consultant, Renaissance Worldwide, 1998-1998. Consultancy. Technical consulting foot-soldiery.

- [Webmaster](#), Minnesota Public Radio, 1995-1998. With the direct assistance of Netscape, created a Web presence for one of the leading public media services in North America. Pioneered early live video netcasts. With [Rivertown Trading](#), pioneered early B2C e-commerce.

Education

Started as a computer geek, then went liberal arts until my wife's own graduate degree in archaeology (Cornell) gave me a dose of reality. Jumped off and joined the Internet revolution.

- Brown University, Providence, Rhode Island. Master's Degree program with fellowship, 1995, incomplete. Archeology and History.
- University of Minnesota. Philosophy.
- Saint Olaf College, Northfield, Minnesota. Bachelor of Arts, Magna Cum Laude, 1989. Double-majored in Philosophy and History.
- Københavns Universitet, Denmark. History. Study abroad while @ Saint Olaf.
- Minnesota State University. Computer Science. Began my post-secondary education as a sophomore in high school.