

Lia Adams, PhD

Technical Writer

Objective

I am a technical writer who worked for years in software research and development before specializing in technical communication. In my technical communication positions, I use my skills in problem solving, invention, observation, documentation, and instruction. I am available for contract work in May 2015.

Experience

My knowledge of computer science and mathematics gives me the vocabulary and experience needed to **observe** and **interview** experts in highly technical subject matter and to **analyze** the documentation needs of engineers developing software and hardware. As a writer of developer documentation, I **organize** and **communicate** technical material in documents, hypertext, and simulations.

In a variety of projects, I have documented software requirements, software designs, implementation plans, and programming interfaces (APIs). I am an inventor on four successful patents.

I learn the language, tools, and culture of new groups quickly.

Education

- Bachelor of Arts in Mathematics & College Scholar at Cornell University
- Ph.D. in Computer Science, Stanford University
- Becoming a Technical Information Developer at UCSC Extension

Work History

2015 - 2016 **Google Inc.** Mountain View, CA

As a contract Senior Technical Writer, I wrote documentation for the API and SDK of the Android for Work App. This app is used by developers of Android apps who need to run their apps on earlier versions of the Android operating system. In this job, I read and documented thousands of lines of source code in Java and interviewed content experts at Google.

2014 Summer **Apple Inc.** Cupertino, CA

Made substantial changes to the iTunes Connect Developer's Guide to improve organization and to include new software features and hardware platforms.

2013 Summer **PayWithMyBank** Redwood City CA

Wrote the *Merchant Integration Guide* for early adopters of the PayWithMyBank payment platform. The guide includes the

- End-user experience for making a payment
- Life cycle of payment transactions
- Overview and examples for merchant integration with the platform
- API Operations
- Data Types

2007 - 2011 **Google Inc.** Mountain View, CA

While employed at Google as a **Senior Technical Writer** I planned, wrote, edited, and updated internal technical documentation.

- **Procedure:** Observed, interviewed, and documented internal procedures for the group of engineers in charge of system reliability of Google Search.
- **Instruction:** Wrote two tutorials and a 100-page user guide for Borgmon, the crucial internal system for monitoring Google services in the field. Borgmon is the fundamental tool for system monitoring, but documentation for it was missing or obsolete before my work.
- **API and SDK:** Documented a new internal library for managing the database of configuration data. Contents included the database definition, API, and a user guide.
- **Analysis:** Analyzed postmortems for outages of external Google services. Involved studying notes from retrospectives on service failures, interviewing engineers involved, and identifying elements common to outages in multiple domains.

2004 - 2006 **Expert Support** Mountain View, CA

For **Sony Computer Entertainment:** wrote, edited, and updated technical documentation.

- Documented prerelease versions of the PlayStation Graphics Library (PSGL) API for game developers targeting the PlayStation3.
- Edited assembly language and C++ manuals for Sony's PPU and SPU processors. Made numerous changes to improve clarity and correctness.
- Wrote 29 Starting Guides to help game developers start writing game software for the PlayStation3.

For **Adobe Systems:** Wrote, edited, and updated internal and external documentation.

- Documented SDKs and APIs for OEMs who develop products based on Acrobat, PostScript, and PhotoShop.
- Did writing and extensive editing on an Adobe mathematical paper describing and PhotoShop's algorithm for repairing damaged images and how it adapts to properties of human vision. This algorithm is the basis for the Healing Brush. Working on the documentation involved interviewing the inventor, and reading the relevant patent.
- Did writing and extensive editing on an Adobe whitepaper "Implementation of Black Point Compensation" explaining PhotoShop's algorithm adapting the level of black in an image for different devices.

2002 - 2003 **Computer Science Department** Stanford, CA

Taught CS 143, an introductory course on the principles and practices of compiler construction. The topics included lexical analysis, theory of parsing, semantic analysis, runtime environments, code generation, and code improvement (optimization). I wrote and delivered lectures, prepared handouts, wrote exams, and supervised the students in their programming assignments, which culminated in a working compiler for a simple object-oriented language. My teaching got rave reviews from students.

2000 - 2001 **Transmeta Corporation** Santa Clara, CA

While employed at Transmeta as a **Technical Communicator**, I contributed to the Instruction Set Architecture (ISA) document for a new processor. Responsibilities included interviewing hardware designers, reading bit layouts of instruction words to describe in English, and scripting the rebuild process for a multi-purpose ISA document that could be used to generate both human-readable text and compilable C code.

1998 - 2000 **Lia Adams Consulting** Palo Alto, CA

As a **Domain Expert** at Reasoning Incorporated, I consulted on the functionality and user interface for a new product. I interviewed internal users, helped formulate the user interface requirements, and documented the implementation plan, including interfaces to the configuration management system, the source-code analyzer, and the database schema.

Contract Technical Writer for Expert Support, purveyor of highly technical writers.

Consultant to complete the prototype of Palette Converter tool at FX Palo Alto Laboratory Inc.

Development Strategist at Anita Borg Institute for Women and Technology where I helped originate the Virtual Development Center.

1996 - 1998 **Fuji Xerox Palo Alto Laboratory Inc.** Palo Alto, CA

As a **Senior Research Scientist** at FXPAL, I co-authored reports and papers, wrote patents, and gave technical workshops at Fuji Xerox Corporate Research Lab in Japan. As the project lead for collaboration through virtual spaces, I planned projects, budgeted and allocated resources, and coordinated project members' activities. I led the team in deploying virtual spaces on the web for collaborative work, using infrastructure from Xerox PARC, The Palace, and PlaceWare.

In adaptive software, I helped create a prototype system for creating programs by manipulating objects. We investigated physical objects not only as end user interface elements, but also as tools for actual programming. To investigate opportunities for tangible interfaces, I observed people giving presentations and analyzed their activities, whereupon our group designed and prototyped the Palette physical interface.

1990 - 1996 **Silicon Graphics Inc.** Mountain View, CA

As one of the first **Members of Technical Staff** on the CaseVision (later ProDev) team, I contributed to the design, implementation, and release of a successful product: the WorkShop software development tool suite, which won the Jolt Award for programmer productivity. During this project, I **interviewed** customers about the software tools they needed, **proposed** and designed features for our programming tools, **negotiated** with other groups in the company to enhance their code to support our tools, **developed** server-side product code in C, C++, and Ada 95, **communicated** with engineers on the WorkShop team to ensure smooth internal integration, **wrote** internal and external documentation, **demonstrated** the products to users, **trained** customers and marketing staff, and **created** automated tests for regression detection.

To integrate various tools and languages including Ada 95, I walked trees, augmented symbol tables, created ToolTalk messages, added queries to the code analyzer and browser, implemented debug-time expression evaluation, and defined an API for definitions and uses of symbols.

1989 - 1990 **Sun Microsystems Inc.** Mountain View, CA

As a **Member of Technical Staff** I worked on *dby*, an advanced client/server debugger, implementing commands in the process control server and extending a *tcl* interface to them. For our compiler front-end engineers, I documented the debugger's API for symbol-table information.

1983 - 1988 **Stanford University** Stanford, CA

For my PhD research in the Computer Science Department, I designed and prototyped the *Ogre* editor for the *grid* (a rich, language-independent program structuring mechanism). *Ogre* allowed programmers interactively to access and modify global structural information within the context of an individual module, while ensuring that program interconnections stayed consistent during program editing. *Ogre's* efficient incremental-update algorithms made interactive *grid* access feasible for real programs, without the support of a monolithic environment.

Summer 1983 **Xerox Palo Alto Research Center** Palo Alto, CA

As a **research intern** on the Voice Project in PARC's Computer Systems Lab, I enhanced the electronic mail system in the Cedar environment to add voice annotations to messages, using database operations to coordinate the management of audio resources.

Other experience at work and school

At **Four-Phase Systems** (now Motorola), I diagnosed and corrected errors in compilers and run-time libraries when field personnel were unable to find the problem.

At **Berol Corporation**, I modified and documented the previously comment-free business applications written in IBM assembly language, and I wrote a 100-page user manual documenting the order-processing and accounts receivable systems.

As a **Cornell** undergrad, I worked as a teaching assistant for classes in programming language principles, numerical analysis, differential equations, and expository writing.

Publications

- Les Nelson, Lia Adams, Satoshi Ichimura, Elin Pedersen: Palette: A Paper Interface for Giving Presentations, in Proceedings of CHI 99, Pittsburgh, Pennsylvania, May 1999.
- Lia Adams, Lori Toomey, Elizabeth Churchill: Distributed Research Teams: Meeting Asynchronously in Virtual Space, in Proceedings of Hawaii International Conference on Systems Sciences, Maui Hawaii, January 1999.
- Lia Adams, Lori Toomey: Designing a Trans-Pacific Virtual Space, in SIGGROUP Bulletin, Volume 19, Number 3, December 1998.
- John Tang, Lori Toomey, Gloria Mark, Lia Adams: Designing Virtual Communities for Work: CSCW '98 Workshop Report, in SIGGROUP Bulletin, Volume 19, Number 3, December 1998.
- Lori Toomey, Lia Adams, Elizabeth Churchill: Meetings in a Virtual Space: Creating a Digital Document, in Proceedings of HICSS, Kohala Hawaii, January 1998.
- Internal FXPAL Technical Reports and Technical Memos 1996-1998.
- Contributed to documents on SGI Developer Magic: *Performance Analyzer User's Guide*, *ProDev WorkShop Overview*, *Static Analyzer User's Guide*, *CaseVision Environment Guide*
- Lia Adams, Stanford University Technical Report CSL-TR-89-376 *Integrating a Program Structuring Mechanism with a Program Editor*

Patents granted

- US Patent 6,509,909 & US Patent 6,195,093 *Systems and Methods for Controlling a Presentation using Physical Objects* by Nelson, Adams, Ichimura, Pedersen & Smoliar describe systems enabling people to control the content and order of a presentation material by manipulating physical representations of that material. Granted 2003 and 2001.
- US Patent 6,175,954 *A System and Method for Creating Computer Programs Using Tangible Interfaces* by Nelson & Adams describes a new way to specify and create computations through manipulating real objects. Granted 2001.
- US Patent 6,119,147 *A Method and System for Computer-Mediated, Multi-Modal, Asynchronous Meetings in a Virtual Space* by Toomey & Adams is a system for conducting and capturing meetings among colleagues separated by time and space. Granted 2000.

Tools

In technical writing, I have used:

- Operating systems: various flavors of Unix and Windows, as well as some MacOS X.

- Markup languages: HTML, XML, Adobe's MML, and a couple of proprietary markup languages.
- Desktop publishing: FrameMaker, Word, Acrobat, PowerPoint, Photoshop.
- All-around tool for doing everything: emacs!
- Doxygen, Javadoc, and other tools for software analysis and documentation in C, C++, and Java.
- Version control: ClearCase, Perforce, Subversion, and CVS.

As an engineer, I used:

- C, C++, some Python and Java, and assembly languages.
- Common Unix scripting languages.
- Various tools for software development, static analysis, version control, and debugging.