

# Warren Scott Dillman

[Click for PDF Version](#)

Website: [dreamcycle.net](http://dreamcycle.net)

home: [sdillman@dreamcycle.net](mailto:sdillman@dreamcycle.net)

## Objective

I am very passionate about writing elegant and efficient code that solves business problems by abstracting the solution in a way that does not suffer from over engineering. I enjoy finding the balance between abstraction and application efficiency, while strictly adhering to the requirements in the problem domain. Producing code that easily conveys the solution to other developers and is maintainable and self documenting--this is the challenge that I enjoy. What I am looking for is a team of like-minded people striving for the same objective, to create an industry leading product by utilizing these principles; and who enjoy the challenges this provides.

## Education

[Northeastern University](#) Boston Massachusetts

Bachelor of Science, June 1993

Concentration: Management Information Systems / Management

[GameInstitute.com](#)

Advanced 3D BSP, PVS and CSG Techniques, September 2001

Concentration: Game Development

[on line certificate](#)

## Experience

IBM Software Lotus Division

*Advisory Software Engineer / Tools Developer*

Release Engineering, December 2003 - Present

I began my time in the Lotus Division supporting and extending the already well established software pipeline originally engineered to build Lotus Notes/Domino on more than ten platforms. The system was built on a suite of in-house tools written in C/C++ by Iris Associates. I was part of a 2 person team responsible for supporting this tool set. My responsibilities included: maintenance of the tools; by fixing bugs and fielding and implementing enhancement requests, as well as documenting and providing education to the development and build staff. Many of these tools dated back to the mid 1980s and gave me the unique experience of troubleshooting many fringe problems introduced by changes in operating environments as well as increased build complexity. Working in this area provided me the opportunity to work with some of the people on the original Lotus development teams who were very generous in sharing their engineering knowledge and experience. This is consider a great personal asset.

Many of Lotus' newer offerings were a mix of Java and native C++ components. Some of the old tried and true tools were still applicable, but because of the large amount of Java; an extended Ant based system was developed. I was asked to take over management of this system in much the same manner in which I managed the older build system. Development of this new build process was very active and allowed me to contribute to both the direction and efficiency of the entire build stack. Balancing the system to provide enough process to the release engineering team to allow them to be efficient and giving enough flexibility to the software engineers to allow them to be creative was my main focus. Giving everyone a common vocabulary in which to share development ideas while still meeting the company's requirements was paramount.

I believe more often than not this balance has been struck, with proof being reflected both in the quality of the product and incredibly large number of daily builds managed by a modestly sized release engineering team. It also has to be said that a large contributing factor to the success of the system was due to the efforts of all parties involved sharing information and working together closely with a common vision to produce the highest quality product possible. This involved groups both local and in many locations, including over seas.

My time at IBM provided me with the opportunity to contribute my skills to the success of a high profile team. It has increased my engineering skills by giving me the autonomy to explore alternate solutions as well as work in a

mentoring environment that provided guidance and knowledge.

Rational Software / IBM Software Rational Division

*Staff Software Engineer*

Web Services, January 2000 - December 2003

I joined Rational Software as the second engineer on the Rational web team. I was fairly familiar with Java as far as client side programming was concerned, but had little to no J2EE experience. Rational immediately sent me to the advanced OOA&D class at Rational University, and it had a huge impact on the way I design my applications. I also attended several Java seminars and was quickly brought up to speed.

I then began work in a J2EE environment as we migrated the Rational.com site from a proprietary app server built on JRun, to Weblogic. It was a difficult and challenging experience, but one of the most rewarding projects I have worked on. Probably the largest team project I have worked on to date and my first experience with the Rational Unified Process. Even the last 2 week crunch period of 14 hour days was a very good learning experience and was a testament to the communication and cohesion of the team.

Since that initial re-architecture most of my work was in the J2EE space supporting the web site as well as adding new features, and re factoring old systems with new designs.

Some of my projects for Rational.com include: Geo-coding the events system to allow customers to search within a zip code radius for company events and classes. Single sign on, to allow Rational Developer Network and Rational.com to share user information. And the porting of the Rational licensing application to Websphere to generate customer evaluation licenses. In completing these projects, I leveraged many Java libraries and constructs, including; EJBs, JSPs, custom tag libraries and other technologies in the J2EE and Java toolbox. As well as making use of commonly understood and reusable patterns.

Most of my contributions were targeted at improving the customer experience and to help the sales force in the field accomplish it's sales goals.

Rational exposed me to many design and development philosophies and allowed me to work along side many people who have helped me grow as a software engineer. There has also been time for many courses in OOA&D , Rational Rose, Clear Case and now WebSphere and WSAD.

Sybase, Inc.

*Sr. Web Application Specialist*

Web Services, July 1997 - December 1999

My responsibilities at Sybase were mostly centered around supporting the web content developers and maintaining the back end systems for the Sybase.com web presence. These areas involved mostly lead capture, and the development of the Sybase Developers Network.

I shared the technical lead position on SDN ( Sybase Developer's Network ) with one other engineer. Together we architected and delivered the site/application built on Sybase's own proprietary web development language called Power Dynamo. It was an important step to show the development community the power of the development system. The site was delivered on time and was a success. Membership steadily increased during my time there, and proved to be a very valuable lead generation tool which helped the sales force in finding new customers.

There were various smaller projects that gave me a chance to continue my C++ and Java development. I was able to write a Windows™ screensaver for the World Cup sponsorship. I successfully designed and implemented a Windows™ CD-ROM autorun UI system that was used on a CD distributed at the annual Sybase user conference. I also developed many in-house tools to automate much of the daily tasks the web content team was responsible for.

My time at Sybase gave me invaluable experience working in the web space as well as allowing me to meet the challenge of being a technical lead on a high profile project. It helped me in improving my communication skills by allowing me to work within a sizeable team of people spread across a large geographical area.

Polaroid Corporation (Independent Consultant )

*In-house Application Developer*

Computational Modeling / Film Imaging Development, September 1993 - July 1997

At Polaroid I was assigned to the computational modeling group as an in-house developer to support the mechanical engineering staff. This gave me the chance to further my understanding of UNIX as well as learn the skills needed to gather requirements and work with internal clients. Most of my work was command line based in UNIX and C and was focused on writing text file translators that were designed to move data between finite element analysis packages and to report on that data. The job gave me a lot of freedom and allowed me to experience a wide range of operating systems and engineering problems. The solutions that were developed gave the engineering staff a way to share information and data between platforms and applications which greatly helped in their effort to support the film development team. This was the ultimate goal of the department.

I eventually left the modeling group and moved to a group responsible for several of Polaroid's film products. My job in this organization was to develop applications to allow people to analyze data collected from experiments done on the film system. The data that was initially collected was being assigned arbitrary numbers visually by different people. I worked with a team to develop a Windows™ based application to identify and quantify the severity of defects in the film system by processing film density information. At the time I used the Borland C++ compiler and the OWL windowing system. The project was a success and actually helped to reduce the #2 reported customer defect in film, by allowing engineers to quickly correlate chemistry changes with defects on a proven scale. The development team consisted of two developers and one business analyst. Interface design was a high priority in order to insure proper use of the application with as little training as possible.

The project taught me a lot about design, user experience, and effective communication in a team environment. It also broadened my knowledge of C++ and common programming constructs and algorithms.

My final and most successful project at Polaroid was a bit smaller in scale, but was much more scientifically intensive. The objective of the project was to save the company money by phasing out a dying array of machines used to report film density with less expensive machines that worked on the principle of sensitometry. The data generated by the new machines was equivalent, but not on the same numerical scale (not unlike the difference between Fahrenheit and Celsius). The new machines also provided additional information that also could be leveraged. It was decided a Windows™ based application would be written to read in the new data format for each film run, process the data, and save it to a database for archiving and reporting. The application had to be very simple to use, reliable, and allow for a high level of configuration.

The team put together for this project was smaller than previous projects. I was the only developer; and was teamed up with one chemical engineer. He would be responsible for describing the calculations the application would be required to make, and validating the data generated.

Most of the configuration that was needed was handled in the GUI and then stored in configuration files. The configuration files could also be modified to support extended features.

We added a simple reporting system that printed a color chart of the film sensitometry information as well as the supporting data, giving the engineers a quick thumbnail to evaluate the quality of the current film production run.

The project, finished in 1997, was a complete success and was still in use until Polaroid ceased film production. It was used to validate the quality of Polaroid's black and white film product as it came off the assembly line. This saved the company hundreds of thousands of dollars and gave them a valuable reporting and archiving system for historical data analysis.

## About Me

---

### Coding/Technology

In my spare time I have contributed minor bits of code here and there to various open source development projects and have become very interested in interactive media development, especially in the area of user experience. The interactive computer experience allows me to combine my musical abilities and artistic interests, it also provides me with a set of unique problems that help further my C++ skills and at the same time give me immediate feedback on my progress. I believe I have gained more knowledge about good OOA&D principles in C++ from the OSS community than almost anywhere else. I am very dedicated to my advancement as a C++ developer. Even though over the last few years my 'on the job' time has been spent in the web/J2EE arena it has helped me progress in other areas I might not have had the time to pursue. My Adobe Photoshop™/graphics and UI skills have definitely benefited from my client side web work, and Java's strict OO design has influenced how I design and implement my code regardless of the language chosen.

In order to learn new technologies I often design applications as sandboxes to better understand how the new technologies work and how they can be leveraged in real world applications. A good example of this is my

exploration into Microsoft's SAPI speech API. I used SAPI to add text to speech capabilities to the WinFrotz2002 interpreter in order to allow visually impaired players to enjoy text adventure games. I picked the WinFrotz interpreter because I have a soft spot for the original Infocom text adventures and introducing them to a completely new audience was an exciting proposition. This project also included the challenge of designing a navigation system for users without the benefit of sight in an environment that is inherently graphical. The application was well received, and still maintains a healthy download rate. Website can be found here: [WinFrotzTTS](#)

When the [Trillian](#) chat client was released, I became interested in it's extension capabilities through it's plugin API. Text-to-Speech seemed like a natural choice and built upon my previous SAPI knowledge. This eventually became the TalkBack [freeware plugin](#). This is my most complete freeware package, as well as being the most complex. In it's development I leveraged many advanced C++ features as well as relied heavily on the STL and the [Boost](#) libraries.

Freeware development is something I like to do to give back to the community that has helped me in many of my personal projects as well as my career as a software engineer. I have an interest in process automation and usually draw the line at doing things that are manual twice, so I end up with a lot of tools to help me accomplish repetitive tasks. The idea being to automate the mundane tasks to free up more time to be creative. I have donated some development time to the [Donationcoder.com](#) community, by offering advice and at times fully coded solutions to problems. During an experiment into Windows™ dialog hooking I entered the GOE 2007 challenge ( Getting Organized Experiment ) in which I developed [CFDButton](#), which extended the Windows™ common file dialog with extended features. The application received favorable write-ups and was even included on a few magazine cover-discs containing other freeware applications.

Most of my freeware, projects, and write ups, can be found [here](#).

## Music

One of my other big interests is music. It has always been part of my life since I was a very young. My father is a luthier, so music has always been in the family. I have been playing the guitar for over 20 years and studied theory in high school and in college. I have also licensed a few pieces of music to companies for corporate and commercial presentations, including Rational Software.

## Cooking

Another of my hobbies is roasting coffee and preparing espresso, but I also very much enjoy cooking. I have two write-ups on my website containing a little of each:

- [Roasting coffee](#)
- [Making pizza dough](#)