|  |  |  |
| --- | --- | --- |
| Jay Brockway  | 118 N Waverly PlaceMount Prospect, IL 60056(847) 778-9068  | JPBrockway@Gmail.comhttp://JayBrockway.us |
| **SUMMARY** | I bring a broad set of technical skills and experience from working on diverse projects in the automotive, medical and financial industries. My software controls earth moving equipment, hemodialysis machines, and calculates the risk on large financial investments. My ability to quickly learn new skills and engineer solutions to problems has made me successful in moving easily between industries. |
| **EDUCATION** | **University of Illinois at Chicago (UIC)***Bachelor of Science in Computer Engineering, December 1998***University of Illinois at Urbana-Champaign (UIUC)****College of Lake County (CLC)***Associate of Science in Pre-Engineering, May 1995, Honors* |

|  |  |  |
| --- | --- | --- |
| **TECHNOLOGY** | **Languages:** | C#, .Net, SQL, LINQ, XAML, HTML, CSS, JavaScript, C, C++, Swift, Java, VB, XML, XSL, Perl, Python, Bash, Ant, NAnt  |
|  | **Operating Systems:** | MacOS, Linux, Unix, Windows, VRTX, VxWorks |
|  | **Software:** | Git, TFS, Octopus, Jira, SQL Server, Visual Studio, Xcode |
|  | **Hardware:** | Oscilloscopes, multi-meters, logic analyzers, in-circuit debuggers/emulators |
| **EXPERIENCE** | **Guggenheim Partners, LLC**, Chicago, IL November 2010 to PresentTechnologies: C#, VB, .NET, SQL, LINQ, TFS, Git, Octopus, Jira, Visual Studio, Crystal Reports, WPF, WinForms, Entity Framework, MVVM, DDD**Director of Software Architecture and Development for Treasury Services Business:*** Architected a new C#/WPF application to replace three VB/WinForm applications created in the early 2000s.
	+ Written as a desktop application to provide the best experience for data entry.
	+ Utilized DDD and MVVM design patterns to ensure the code is modular, readable, maintainable and extensible.
	+ Organized core “plumbing” into base classes and extensions then created Visual Studio templates to simplify development for junior developers.
	+ New application initially hosted the WinForms from the original applications to provide a seamless transition for the users and avoid supporting multiple applications.
	+ Developed an inter-application notification system utilizing RabbitMQ. A window in the new app displays notifications for completed reports and actionable changes such as new trades. This helps users work more connected as if they were in the office.
	+ Built-in a document system using a rich text editor so that users can write documentation for each page in the application.
	+ Added user preferences to customize the application and choose favorite reports for quick access.
	+ Developed solutions to missing features in WPF such as:
		- the ability to have more than one database table in an editable GridControl.
		- moving data validation from the ViewModel to the Repository to maintain separation of concerns.
		- an XAML extension to overlay two icons for buttons.
	+ Architected a new report system:
		- Interfaces enforce that each report is defined by a single class containing all knowledge about the report including user parameters, database query and layout.
		- Communication with the main application is by messages to maintain separation.
		- WPF controls for report parameters are programatically generated based on parameters each report requires.
		- Allow for multiple report types such as Crystal Reports and Excel.
		- Run reports in the background.
* Developed an automated build, test and deployment processes to streamline our workflow and avoid process errors during production deployments.
* Created a Blue/Green deployment process and an application to load the Blue or Green version of the application. This allowed us to deploy changes the business needed during business hours when all resources are available.
* Eliminated production issues by performing strict code checks for indeterminate and fragile code.
* Developed a program to inventory the stored procedures used in our 140+ Crystal reports to know which reports were affected by database changes.
* Developed a program to list the hierarchal relationships between SQL stored procedures, functions and views. This tool illuminated which reports would be affected by a change in a low-level database function.
* Developed a program to test our reports to ensure changes to the database didn’t affect user reports. This test automation saved hours of manual work.
* Moved the database into a Visual Studio database project and fixed hundreds of issues preventing the project from building. Database changes now follow the same versioning and deployment process as the code.
* Cleaned up the database by removing hundreds of unused functions, stored procedures, views, triggers, tables, etc. which had accumulated over a decade. With only relevant database objects remaining, our understanding and ability to refactor the database increased.
* Combined two databases into one to eliminated a significant amount of SQL code and improve performance due to the elimination of cross database queries.
* Performed several major refactorings of the database schema and data to align with changes in business practices.
* Fostered a symbiotic relationship with the Managing Director of Operations to analyze the data using business rules and present anomalies to data owners for correction.

**TFS Administrator 2010-2019:*** Orchestrated software and hardware upgrades.
* Added, merged and diverged TFS collections and projects.
* Maintained system and user security by utilizing my TFS administrative tool which lists user permissions in all collections.
* Provided user training and assistance.
* Fostered an Agile development process by introducing Git and creating Kanban boards.
* Developed a TFS build task tool [[Set Assembly Version](https://marketplace.visualstudio.com/items?itemName=jaybrockway.SetAssemblyVersion)] to facilitate tracking of software changes from source code to executable.
* Developed a corporate wide deployment tool [Patchman]. Features include:
	+ A plugin architecture using the Factory pattern makes the addition of new functionality (tasks) as simple as creating a new class.
	+ An XML based scripting language syntax similar to MSBuild and NAnt to allow for more complex deployment processes including the use of variables.
	+ Detailed deployment per auditor requirements.
	+ Detailed error information to indicate source of failure and simplify debugging.
	+ Fully documented user manual with examples.

**Octopus Administrator 2017:*** Installed, configured and demoed deployment process for corporate wide use.
* Helped users with setting up their projects and installing tentacles.
* Modified Patchman into an Octopus step template.

**Jira Administrator 2016-2017:*** Installed, configured and administered users.
* Oversaw migration of data from Gemini system.
* Created a process within Jira for users to approve changes and a report showing who performed the approvals. This made it easy for auditors to verify the software development process was followed and met their approval as proof of Statement on Standards for Attestation Engagements (SSAE) No. 16, Reporting on Controls at a Service Organization.

**JPMorgan Chase**, Chicago, IL March 2008 to November 2010Technologies: NAnt, C#, SQL, ClearCase, TFS, Visual Studio, MSBuild, CruiseControlReason for Leaving: Career growth in a development and architect position.**Lead Deployment Engineer:*** Configured and deployed large-scale web applications to a Windows server farm.
* Saved the company thousands of dollars per deployment. My first deployment was a 48-hour marathon beginning Friday evening. 90 people rotated on site. 20 hotel rooms were purchased for naps between shifts. In contrast, my last and largest deployment was an 18-hour event starting Friday evening with a staff of 30, some on site and some remote. No hotels were needed. No incidents occurred and the deployment was declared a success on noon Saturday. After my first deploy experience, my goal was to tackle the complicated and error prone deployment practices.
* Created NAnt scripts to automate and enhance robustness of the build and deployment process.
	+ Added error handling to recover from non-fatal problems such as creating missing directories.
	+ Performed system testing to detect potential server configuration issues such as testing network connections and testing for required components like .NET.
	+ Verified production passwords against LDAP prior to the start of deployment thus potentially avoiding hours of re-deployment.
	+ Eliminated manual steps.
* Built a GUI to call deployment scripts.
	+ The GUI is configured via a built-in XML document. This interface was adapted to new applications by other teams.
	+ GUI presented different deployment options based on DEV, UAT, and PROD environments.
	+ GUI eliminated errors from constructing complicated command line parameters in the command window.
	+ The simple interface made it easy for production administrators and new hires to perform deploys with little training.
* Developed custom NAnt and MSBuild tasks using C#.
* Excel runbooks were used to keep track of each step of the multi-department deployment process. I created VBA macros to automate and validate changes made to the worksheet. The goal was to avoid mistakes while performing overnight deployments.
* Mentored co-workers in technologies and application architecture.
* Researched and debugged application failures, performance bottlenecks, and memory consumption. These activities were not part of the deployment team job description but my engineer inquisitiveness sparked my curiosity.
	+ Uncovered problems plaguing the performance and reliability of the applications such as remoting exceptions, zombie services, memory leaks and network slowness.
	+ Provided solutions to application architects, developers and support teams.

**Wachovia Bank**, Charlotte, NC (transferred to Rosemont, IL) November 2004 to January 2008Technologies: Java, Ant, Perl, C#, .NET, SQL, VB6, VBA, Crystal Reports, HTML, CSS, XML, XSL, JavaScript, AJAX, Unix shell scripts, C++, CVS, PhotoShop, Visual StudioReason for Leaving: Position cut due to staff reduction. Offered a new position requiring a move to Charlotte, NC.**Software Engineer, GUI Designer – Financial Application Development – Assistant Vice President:*** Received outstanding service award for my work on the Market Risk Explorer application.
* Implemented a multi-threaded Java process to concurrently read data from both a Sybase and MS SQL server database simultaneously to compare the data for discrepancies and generate an Excel report.
* I enabled the bank to reinvest $1.5 billion dollars by completely rewriting their unreliable financial investment risk calculator into a stable and efficient application. Cleanup included
	+ Trimming application out of a much larger inoperative application
	+ Removing J2EE
	+ Designing a database connection class
	+ Properly handling exceptions and exit statuses
	+ Integrating non-Java pieces.
* Rewrote an Excel macro that calculates beta and volatility as a Java application.
* Developed Java JDBC database access utility class to provide commonly used functionality not included in the latest Java JDK releases such as connection pools and row count.
* Developed a SQL Query client in Java to give Unix shell scripts access to MS SQL Server running on Windows.
* Developed complex ANT scripts to build and deploy financial applications to a server farm. File versions were stored in an XML file that acted like the configuration record in Clear Case. The specified version of each file was then retrieved from Visual Source Safe.
* Developed a Perl application to read/write encrypt/decrypt database password and configuration settings stored in XML files.
* Supported the Global Risk System (GRS) financial investment risk calculator by investigating trade processing failures and sending reports to individuals responsible for erroneous data.
* Created with Perl and SQL a fault tolerant, multi-threaded and load distributed report daemon (bot), which spawned the creation of over 3000 daily reports. Status of reports were displayed in a terminal window and a web page.
* Developed with Perl, SQL and JavaScript a project management web application to track feature and bug request development.
* Designed with Perl an efficient algorithm to rapidly determine the differences between data sets from Visual Source Safe 6 and a SQL Server database.
* Introduced CVS software repository to our group and was group librarian.
* Started our group’s intranet site to facilitate knowledge sharing.
* Created Risk Management reports using XML, XSL, and CSS that resembled their original Excel version.
* Developed with C#, .Net, AJAX, CSS, JavaScript, and SQL (tables, stored procedures and triggers) a web application dashboard to display the performance of monitored financial applications. Features included
	+ Process hierarchy tree.
	+ Trend graphs.
	+ Performance tables with a heat index indicating success/failure rate.
	+ Comment and issue dialog boxes to enter information by support staff.
	+ Fully administered by web.
	+ Highly flexible.
	+ Fully normalized SQL tables.
	+ Data security integrated into stored procedures.
	+ SQL triggers enforce table relationships.
* Developed custom JavaScript AJAX library. Methods included
	+ Handle asynchronous connections with SOAP server.
	+ Populate drop down lists.
	+ Populate tables.
	+ Clean data.
	+ Verify form data.
* Developed an Excel macro to compare data rows in a selected column.
* User interface architect and designer. Created layout and icons for all user interfaces designed in our group.
* Developed a web-based SQL client application for remote administration.

**Caterpillar**, Peoria, IL April 2003 to November 2004Technologies: C, Assembly, Make, VBA, Unix shell scripts, Perl, ClearCaseReason for Leaving: 12 months contract through Technisource.**Computer Engineer:*** I saved the company from spending four-hundred thousand dollars in new equipment to reduce compilation times in our remote office by setting up a network tunnel to the central office that enabled remote engineers the ability to quickly build their code on the main office servers.
* Redesigned the software build process to cut build times from thirty to fifteen minutes therefore saving hundreds of dollars in man-hours per day in idle time.
* Developed a tool in C that hooks into the pOSEK operating system to analyze the performance of system processes in real-time. Features included
	+ Easy viewing of performance data via an in-house Windows application.
	+ Easy integration as a library.
	+ High configurability allowing engineers to specify processes and code blocks to watch.
	+ Statistics such as time slice slips, process minimum, average and maximum run-times, number of times process runs.
	+ Statistic logging of running processes used to trace cause of computer resets due to CPU overload.
	+ Detailed documentation specifying usage, performance computation algorithms, and limitations.
	+ Insignificant CPU clock cycle overhead.
	+ Highly accurate time calculations.
* Developed software in C to control an anti-slip differential which routed torque from the wheels that slip to the wheels with grip. Algorithm based torque vectoring on wheel speed and articulation angle of vehicle.
* Developed, cross-compiled, and test embedded applications, which run under the pOSEK, VRTX and VxWorks real-time operating system.
* Integrate, compile, test, and release several software libraries and templates.
* Created a Perl tool to parse source code and generate a metrics report.

**Siemens Medical Systems (contractor)**, Hoffman Estates, IL December 2002 to April 2003Technologies: C++, ClearCaseReason for Leaving: 4 months contract through R2 Staffing.**Software Engineer:*** Designed and developed in C++ an application to test the video graphics processing system for a 3D angiograph machine.
* Wrote detailed software design documents.
* Archived software revisions using ClearCase version control tools.

**Aksys Ltd.**, Lincolnshire, IL January 2001 to December 2001Technologies: C++, TCL, Make, SQL, Unix, CVSReason for Leaving: Engineering staff cut after product completion due to financial difficulties. **Software Engineer:*** Developed, compiled, and tested C++ software for a home dialysis machine running the QNX real-time operating system.
* Designed and developed the scheduler subsystem in C++. Functions included
	+ Monitor the patient’s prescription, treatment schedule, and blood and dialysate test schedule.
	+ Run processes to prepare machine for scheduled treatment.
	+ Monitor the systems cleaning and sterilization process.
	+ Monitor blood filter performance.
	+ Monitor age of dialysate.
	+ Execute the necessary tasks at the proper time to prepare for treatment.
* Designed and developed test software and testing procedures to verify scheduler subsystem's integrity.
* Created and wrote detailed software design and test procedure documents following FDA guidelines.
* Began designing and prototyping a self-diagnostic tool that alerted the service team of necessary maintenance via email.
* Assisted in the verification and validation of the home dialysis machine for FDA approval.
* Verified and validated software applications.
* Reviewed source code for correctness and proper coding standards.
* Stored and retrieved system variables using stored SQL commands.
* Used inter-process communication (IPC) to communicate with other processes.
* Saved software revisions using CVS and Perforce.

**Caterpillar**, Peoria, IL April 1999 to January 2001Technologies: C, Make, Unix shell scripts, ClearCaseReason for Leaving: 12 months contract through Technisource. **Computer Engineer:*** Developed, cross-compiled, and tested embedded C applications in Unix, which ran under the VRTX real-time operating system.
* Designed and developed in C and assembly a set of hardware abstraction libraries (HAL) to provide the application developers a hardware independent API.
* Designed and developed an initialization routine to set up a computer control module in a known ready state.
* Designed and developed embedded C applications to test every circuit of an electronic control module prior to shipping.
* Created and wrote detailed software design, API, and test procedure documents following ISO9001 guidelines.
* Assisted in the trouble shooting and set up of a new production line.
* Resolved production line testing issues.
* Utilized schematics, oscilloscopes, logic analyzers, in circuit emulators, and custom setups to perform extensive testing of computer control module software and hardware functionality.
* Helped the hardware developers solve circuit design problems.
* Reviewed source code for correctness and proper coding standards.
* Wrote hardware trap routines.
* Used J1939, CAN, and CAT protocols for inter module communication.
* Stored software revisions using ClearCase version control software.

**NetDox**, Deerfield, IL June 1998 to April 1999Technologies: Unix shell scripts, UnixReason for Leaving: Company closed.**Junior Unix Administrator:*** Monitored and tested network functionality.
* Monitored all computer and network hardware with HP OpenView SNMP.
* Monitored production system processes and workload.
* Monitored Sun tape library and backup process.
* Resolved and corrected system faults.
* Created and documented procedures.
* Configured workstation and production machine hardware.
* Installed and maintained Solaris workstations.
 |
| **TRAINING** | * Digital Systems
* Digital Networks
* Numerical Analysis
* Discrete Mathematics
* Computer Architecture
* Computer Systems Design
* Microprocessor-Based Design
* Data Structures and Algorithms
* Computer Communication Networks
* Image Analysis and Machine Vision
* Operating Systems Concepts and Design
* Principles of Real-Time Systems
* Introduction to Rational ClearCase
* Computer Organization and Programming
* Object-Oriented Analysis and Design with Unified Modeling Language
 |