Craig Spry

Australia
Email:craig.spry@gmail.com

Experience Summary

I have been a software engineer for over twenty years. I have a wide range of experiences, programming everything from microcontrollers to server applications and have proven myself capable of delivering solutions on multiple occasions.

Technical Summary

Languages	Python, Java, Golang, R,Javascript, C++, C, C#, PHP
Web Technologies	Django, HTML, CSS, REST, Cometd, Wordpress, Drupal, XML, XSLT
Blockchains	Ethereum, Nem
IDE	PyCharm, Visual Studio(4.2-2010,Code), Eclipse, Netbeans, IntelliJ, RStudio
Source Control	GIT, Hg, SVN, CVS, VSS
Web-servers	Tomcat, Apache, IIS
Operating Systems	Windows(XP/2003/2008/20012/7/10), Linux, OSX,Freedos
Java Libraries	Struts, Tapestry, Spring, Tacos, Axis, Apache-XMLRPC
Javascript Libraries	AngularJS,JQuery,RGraph
Continuous Integration	CircleCI
AWS Technologies	IAM, SQS, Elastic Beanstalk, Light sail, RDS, EC2
RDMS	PostgreSQL,SQLite,Oracle
Build Tools	Ant, Maven
Reporting tools	Jasper Reports, Jupyter

Education

Victoria University of Technology

1994-1996

- BA, Applied Physics and Computing

Experience

Acusensus

May 2018-Present

Acusensus is a traffic safety organisation focused on providing systems to positively modify driver behaviour C#, SQLite, Java, AWS, C++. Contractor

- Formulated incident file format, with encryption.
- Built incident extractors in Java and C#.
- Built incident viewer.
- Integrated automated number plate recognition(ANPR) into the camera solution
- Built a suite of tools to quickly test ANPR configurations
- Secured AWS

Adcoin Technologies

August 2015-Present

Adcoin Technologies is a digital advertising service that monetizes attention. Utilising Python, Django, Postgres, Redis, CircleCI, AWS. Chief Technology Officer.

- Implemented initial ad serving logic.
- Maintained ad editing interface, used by advertisers to construct ads.
- Built the system to keep track of all the financial transactions for paying publishers and accepting payments from advertisers.
- Built a zero point paywall so publishers could use Adcoin advertisements as their paywall.
- Implemented a tracking pixel.
- Configured and maintained the AWS/CircleCI environment.
- Provided technical support and advice for publishers and advertisers to integrate with Adcoin
- Maintained the javascript snippet that allowed for a copy and paste integration.
- Assessed various blockchain technologies.

Solar Systems

April 2012 - July 2015

Solar Systems designed and manufactured high efficiency photovoltaic power generation systems. Software Engineer.

- Developed a site journal where activities that were performed at the power stations could be logged, using Golang, JQuery and Postgres. This was a small web application to facilitate the logging and review of operations and maintenance activities. These would then be exported and imported into a back office system.
- Prototyped an embedded controller based on the FRDM-K64F board. I ported the FreeMODBUS library to FreeRTOS and lwIP and ported the existing SNTP client to work on the FRDM-K64F.

Additionally I worked on motion control for tracking the sun.

- Implemented a Postgres database to store data coming in from the remote power stations. The files were transferred in as CSV from the on site logging system. On top of this I used JasperReports to generate the reports.
- Used R and RStudio to perform analysis on the data in the above mentioned database and long term solar resource data in CSV files. This analysis was used to aid the design process for the next generation dish and to track site performance.
- Worked as part of a team that added a closed loop transfer function to the sun tracking code that runs on the dish systems, ABB AC500 PM573-ETH V2.1. The method that was previously used to track the sun had trouble tracking the sun at high angles due to structural deformation. The closed loop solution that we implemented overcame this problem. I worked with a controls expert who came up with the transfer function and translated this to Structured Text.
- Implemented a JasperReports server to serve up the reports for the database. As part of this I used LDAP to make the server single sign on so the users could use their existing password to log on.
- Dashboard for showing site status. Using Golang, JQuery,AngularJS, RGraph and Postgres. This is a small web application to show the status of power stations on a large screen.
- Implemented Mercurial version control, Mantis bug tracker and an internal instance of MediaWiki. When I first arrived at Solar Systems I was taken aback by the lack of version control, bug tracking and a wiki. So I installed the aforementioned applications.
- Built a dish simulator in Python. So I could run a simulated plant I implemented a dish simulator, this would track just as a dish would and could be used to simulate alarms and events just as they would happen on site.

Redflex Traffic Systems

April 2001 - 2012
Redflex Traffic Systems is a provider of automated traffic enforcement solutions.
Software Developer - Team Leader

- Developed Back office processing system for infringements, using Java, C#, Tomcat and Postgres. I was initially involved with the development of a prototype, then I was the project manager when this was being customised for a customer. This involved interfacing with the clients system so look ups on plate numbers could be performed, using an ActiveX control. To deliver this I spent a great deal of time working on the client's site.(Java, Struts, C#, Postgres, ActiveX)
- Developed speed/redlight camera software. I was a developer on this project using Microsoft Visual Studio, MFC and COM. I was responsible for developing the interfaces to the speed measurement devices, weigh in motion devices. I also developed a light meter interface, the configuration front end and the business logic for capturing vehicles, the number of shots to be taken and when to take them. Developed the interface to various variable speed limit signs and configurable business logic of how these would interact with a camera. For this project I spent some time on site testing.(MFC, DCOM, ATL, Visual C++)
- Maintained the code for a speed measurement device. This system ran on an embedded 386 running free dos. The code for this was written in Borland Turbo C++.(Freedos, Borland Turbo C++)
- Developed speed measurement device using a Motorola 68HC08 micro controller. I was part of a team of two people developing this device, I did the software and some of the hardware and the other person did most of the hardware and some of the software. To do this C was used with a

little assembler using Codewarrior as the IDE.(C, Assembler, Codewarrior)

- Developed processing system to capture vehicles travelling between points. I was the developer/team leader for this project. This project used Postgres, Java, Tomcat, C++. I was involved in developing the vehicle matching algorithm, building the user interface, optimising the DB.(Java, Tapestry, Tomcat, C++)
- Developed a system to detect non buses traveling in bus lanes. This is a variation of the point to point system and used the same technology. I was involved with the development, installation and acceptance testing of a number of these systems.(Java, Tapestry, Tomcat, C++)
- Developed a system to monitor large numbers of digital cameras. This system downloads files from and monitors the state of digital cameras connected to a central location. This was done using Java, Tapestry, Tomcat and Postgres. I was the team lead on this project and was involved in the selection of technologies and development of the user interfaces and the interface to the cameras and optimising the system to get the response times required by the customers. (Postgres, Java, Tapestry, Tomcat, CometD, Web Services)
- Worked on site at Redflex Traffic Systems Inc, Scottsdale, Arizona, USA helping of implement back office system for incident processing for various cities that Redflex had the Redlight/Speed processing contracts for. (SQLNavigator, Oracle)
- Developed a web interface for controlling and configuring a camera. I was involved in the initial prototyping of this and the early stages of development. PHP, javascript and ajax are used for this. This also involved writing an XML-RPC interface to the camera software.(AJAX, PHP, XML-RPC, C, Apache, Apache Modules)
- As well as being involved with the development of the above systems I was also involved in writing the user manuals and conducted training courses.

Neumaflo Australia

April 1997 - March 2001
Software Developer

- Part of a team that developed a case tool for PLC. This was developed using Microsoft Visual Studio and MFC. This product had a graphical user interface and for editing sequential flow charts and ladder logic and introduced the concept of Object Oriented Programing to PLC programmers.
- Developed auto document generating component. This involved turning the ladder logic and sequential flow charts into English, it was presented this in HTML. As part of this an Internet Explorer component was integrated into the product for displaying the documentation as the user edited the diagrams.
- Developed code generators to turn above diagrams into code for multiple targets, Allen Bradely, Omron, Foxborough and Siemens. For this I made a common text output which was then processed into the format required by each vendors tools. The post processing was done by script files so new PLCs could be easily implemented.

(Visual C++ 4.2/6.0, MFC, Perl, HTML)

Open Source

Tacos(http://tacos.sourceforge.net/tacos4.1/)

- Contributed Comet component. For this I married together the Comet client that is part of the Dojo library and the Comet server that is part of the Jetty library.(Java, Tapestry)

Websites

- http://www.spry.net.au This is my personal website used a pointer to all my other works scattered around the web. This is currently a static HTML5 site that uses AngularJS.
- https://bitbucket.org/craigspry This is where I store my various bits and pieces that I would like to share with the world, mainly Arduino projects with some Python and an Android app.

Interests

- Astronomy/Astrophotography.
- Arduino