David Corcoran

8085 SE Palisades Ln.

Port Orchard,Washington

david.corcoran@live.com

http://krch.freeshell.net/dcorcoran

General Development Areas

System Programming	Application Programming	Programming Tools
System Administration	Embedded Systems	Software QA
Technical Documentation	Hardware Test Engineering	Scripting
Database Programming	Open Source Technologies	Customer Support

Systems/Languages/Protocols/Utilities

Linux, Unix, AWS, VxWorks, Windows (XP, NT, 7, 10), RT/11, RSX/11M

perl (Win32::OLE, Win32::ODBC, DBI, CGI, LWP, LDAP, XPath, HTML::TreeBuilder, VM::EC2)

shell, ruby, C/C++, Oracle PL/SQL, mysql, SQLite, Postgresql, TCL/TK, Javascript, jQuery, CSS, Powershell, Assembly Language for various processors

make, sed, awk, lex, yacc, m4, git, github, cron, jira, Perforce, MS Office, P3e

HTML, XML, JSON, TCP/IP, SMTP, FTP, MIME, HTTP, NFS, NNTP, NIS, AD, wireshark, DOT (graphviz), LaTeX

Employment History

Sep 2016 – Oct 2017   **Software Engineer**   **Kwikee† – Peoria Ill.‡**

Wrote automation scripts, in perl, replacing manual image conversion processes. Provided bugfixes for existing code.

Dec 2015 – Present (concurrent)   **Independent Consultant**   **Port Orchard**

Providing IT support for local attorneys; current projects include setup of Wordpress blog, website maintenance, document preparation and typesetting

Sep 2015 – Dec 2015   **Software Engineer**   **Wells Fargo† – N.C.‡**

Wrote powershell scripts to automate outlook email and putty terminal creation.

Sep 2014 – Aug 2015   **Software Engineer**   **F5 – Seattle**

Modified perl/cgi web pages: decreased load times, moved some functionality (sorting searching etc) to client making use of sundry jQuery libraries. Wrote tools to ease use of somewhat cumbersome utilities, automate AWS-related tasks for VM reporting and cleanup. Fixed bugs, made feature enhancements.

Dec 2013 – Jun 2014   **Build Engineer**   **Revolution Analytics† – Seattle**

Created reporting website for tracking of Jenkins builds. Wrote package blacklist-er script in ruby utilizing SQLite that examines package dependencies by scraping the CRAN website, and producing a list of packages unsuitable for commercial use, based on the license under which a given package is published. Wrote package converter in python that converts CRAN packages to RPM to be used by Terradata PUT utility for package installation.

Feb 2013 – Sep 2013   **DevOps/Build Engineer**   **Disney† – Seattle**

Wrote build scripts in perl to automate the production of static html pages from Visual Studio sources. Migrated perforce projects for United Messaging to git to build under Jenkins. Had to work around a corruption in the perforce database but was able to salvage history past the point of corruption.

Sep 2009 – Feb 2013   **Build/QA Engineer**   **Amazon – Seattle**

Responsible for builds and deployments, for Amazon's FBA suite of services composed of some 174 discreet services as well as the automation of outbound services tests. Wrote numerous database driven perl-cgi pages querying various Amazon systems utilizing multiple mechanisms (RESTful/XML, db, cli) to:

- Create nightly build requests, monitor and report build progress
- Track outstanding changes in FBA packages in perforce
- Monitor deployments to provide developers a summary snapshot status of changes of interest
- Interrogate build system for package information, used as a starting point for code coverage examination
- Drive test data generation data structure creation to simulate the creation of orders, querying catalog for product attributes
- Submit fake inventory for test purposes

May 2009 – Aug 2009   **Customer Support**   **Plainblack – Madison Wis.‡**

Provided technical support for WebGUI, an open source Application Framework, as well as system administration tasks, remotely, in a Linux environment

---

† - Contract (agency)

‡ - Remote

Jul 2008 – Dec 2008　　　　　　　**Perl Developer**　　　　　　　**Courttrax - Bellevue**

Designed, developed and implemented database backend for courttrax.com's data feeds from the Alaskan state court system for web presentation. Also wrote application to scrape Washington State BAR Association website and populate an internal database used in Courttrax's court data aggregation service.

Set up monitors to alert staff to revision changes on PACER web site. Responsible for bug fixes of code that 'broke' due to changes in external websites.

Mar 2008 – Jun 2008　　　　　　　**QA Automation Engineer**　　　　　**Attachmate† – Seattle**

Wrote tests for Attachmates's RSIT for Unix ssh client and server.

May 2006 – Mar 2008　　　　　**QA Test Automation Engineer**　　　　**Isilon Systems - Seattle**

Responsible for the development of automated test written in perl for the test of Isilon's OneFs file/operating system. The tests covered things such as file timestamps, permissions, ownership, and filesystem events, network isolation between the 'backend' NIC with respect to the 'frontend' (wrote a perl wrapper for tcpdump as nmap was not available on the platform under test).

Converted DOT program output, meant to display state bubbles, into perl data structures representing hierarchy of the decision logic for buffered-writes and then analyze these relationships to determine if the correct decision paths were being taken.

Nov 2005 – May 2006　　　　　　　**Build Engineer**　　　　　　　**Pixelworks† – Seattle**

Responsible for the creation of makefiles, shell/perl scripts for the production of CDs, in a Linux environment.

Aug 2005 – Jan 2006　　　　　　　**Perl Programmer**　　　　　**Fred Hutchinson CRC† – Seattle**

Designed perl script to determine changes made in PeopleSoft personnel database and propagate those changes to AD to enable single-source, Enterprise wide control of Contacts in the Address Book and mailboxes in Exchange as well as the various unix systems

Oct 2004 – Jul 2005　　　　　　　**SDET/Builder**　　　　　　　**Microsoft† – Redmond**

Wrote perl modules for the build framework to be used to build Microsoft's Live Meeting Communicator.

Converted batch files for the regression testing of Microsoft's Security Configuration Wizard to jobs to run under WTT, Microsoft's Windows Test Technology infrastructure; this entailed developing perl scripts to add and schedule jobs in WTT

Nov 2003 – Oct 2004　　　　　　　**Partner Support**　　　　　　　**Qpass - Seattle**

Responsible for the support of wireless carriers with respect to software updates and new merchant product integration

Wrote Win32::ODBC scripts to extract merchant data to provide to wireless carriers for maintenance and reporting purposes

Dec 2002 – Nov 2003　　　　　　　**Customer Support**　　　　　　　**Microsoft† – Issaquah**

Provided Technical support for Microsoft's Services for Unix product that incorporated:

- writing working code samples to demonstrate usage of various SFU APIs
- research, documentation and working code sample generation to provide solutions to customers' interface issues between Windows and Unix (issues that required mapping a given Unix process to Windows)
- troubleshooting customer issues, including network trace analysis of NFS/SMB traffic.

The customers I supported were developers, system and network administrators that had in-depth understanding of systems they managed.

The major components in SFU, a POSIX subsystem that sits at the same level as the Win32 API, include Server for NFS, User Name Mapping, Client for NFS, Gateway for NFS and Server for NIS.

Jul 2000 – Aug 2002　　　　　　　**System Programmer**　　　　　　　**Boeing† – Renton**

Wrote perl (DBI) and PL/SQL scripts to extract data from the various Boeing databases to load Primavera Project Planner (P3e) tables

Developed shell and perl scripts under HP/UX to process periodic feeds from legacy systems

Developed client/server utility under Win32 to perform remote table loading across Win2000 and Unix platforms.

Developed Web scraping script, using LWP, to extract manufacturing data where no plain text feed was available.

Mar 2000 – Jul 2000　　　　　　　**Perl Programmer**　　　　　　　**ATL† – Bothell**

Wrote perl scripts to extract function information from C sources to be written as XML for use by an Ultra Sound Controller state machine

Oct 1999 – Jan 2000　　　　　　　**QA Lead**　　　　　　　**Real Networks† – Seattle**

Wrote various utilities in perl and shell to reduce the manual intervention in the software build/release process

Wrote test cases for, and bug entries againt Real Networks Broadcasting Network Product.

---

† - Contract (agency)

Oct 1998 – Oct 1999   **Test Automation Engineer**   **Intel† – Dupont**

Wrote Tcl/Tk scripts for the test of Server Management Firmware for Intel's High End Server Division.

Automated manual tests, reducing the test time in two cases by as much as 98%. Modularized the Sensor Test originally written for one platform to run on others merely by specifying a different Sensor Record file.

Oct 1997 – Oct 1998   **Test Automation Engineer**   **Boeing† – Renton**

Wrote Perl scripts to process data feeds exported from legacy systems to import to MS SQL Server, including an interface to SMTP, for the purpose of issuing email notifications in the event of a failure, and a Web site to cross-reference and catalog the results.

Jun 1997 – Oct 1997   **System Administrator**   **Real Networks† – Seattle**

Wrote concurrency post http log processing utility in C++ & Perl.

Wrote site checking utility in Perl to "listen" to servers and send "trouble" pages.

Debugged and installed Amanda, University of Maryland's Network Disk Archiver, to backup servers over the Internet.

Sep 1995 – Apr 1997   **Firmware Engineer**   **Sensory Inc – Port Orchard**

Responsible for development of real-time firmware, written in assembly, for Voice Recognition and Synthesis consumer products. Also developed and maintained tools and utilities to facilitate the group effort of code development.

Created suite of structured assembly language macros; they brought inline looping control, if/else clauses and case statements to the assembler, which greatly facilitated code development and maintenance.

Developed sophisticated make files to automate the generation of include files which disencumbered the programmer from manually declaring PUBLIC and EXTERN symbols in the source files. Others in the group attested that this saved them many hours and headaches during the life of a project.

Wrote source code cloaker in perl which substituted descriptive symbols with non-descriptive so as to cloak, or hide, their usage and meaning from the users of Sensory's developer's kit.

Integrated Codewright, Opus make and Source Safe to produce a single user interface to the code development cycle.

Nov 1994 – Sep 1995   **Software Test Engineer**   **Microsoft† – Redmond**

Responsible for the test of Data Center back-end data collection task for the Microsoft Network (MSN) on-line service. Developed utilities, generated test cases and wrote bug reports.

Developed test plan to drop data files, at a fixed rate with known good data, to stress the Event Collector and its associated data base. Also wrote the required utility in C++ to convert ASCII data to Binary log files, and the perl scripts to drive it.

Developed various perl test scripts to notify key personnel in the event of a Event Collector malfunction which greatly helped facilitate fault isolation.

Wrote perl lexical scan utility to perform a sanity check of ASCII data issued from data center.

Automated many manual tests using perl and thus increased coverage of regression tests.

Sep 1988 – Nov 1994   **Hardware Test Engineer**   **Aspect Telecommunications – San Jose Ca.**

Worked closely with engineering to specify tests, develop and maintain test software/fixtures and write test procedures for the test of telecommunication equipment. Provided system administration services for the UNIX boxes on the manufacturing floor, including network troubleshooting, system configuration and generation of bootstrappable tapes for software installation.

Developed diagnostics for VME based, embedded processor (68010, 8051), voice cards in C.

Wrote test equipment and low level communication interface routines for use by automated test programs.

Developed and maintained programming environment for test software development.

Wrote T1 bypass diagnostics in C which were highly acclaimed by production personnel.

Extracted production software from a UNIX environment and ported to VxWorks enabling code re-use thus reducing development time. This task required deft use of many UNIX utilities: nm, awk, sed, cxref, ctags etc.

Developed Windows 3.1 based peripheral test eliminating the need for production inventory ($300K).

Wrote many ksh/sh utilities to aid in code development and management of the hardware test systems in manufacturing, including (without limitation):

1. Extraction of file dependencies (*.h) file from C sources Extraction of environment variable names from makefiles to aid in their maintenance
2. Generation of tags files from cxref output enabling access to any symbol in a file
3. Extraction of function prototypes from C sources for human as well as compiler consumption
4. Scripts to control revision and delivery of files to networked test stations
5. Distribution tape generation and backup utilities

Jan 1988 – Jun 1988   **Firmware Engineer**   **Qume - Milpitas Ca.**

As a part of a seven member team, developed firmware for a ASCII terminal product.

Wrote utilities to ease pain of system management on HP 64000 emulator system.

Designed and implemented keyboard driver.

Developed suite of HP 640000 assembly macros to generate scan code tables easily.

Oct 1981 – Jan 1988   **Associate Test Engineer**   **Perkin Elmer – Hayward Ca.**

Provided general trouble shooting support and system management of PDP/11s, under RSX11, in the manufacturing area.

Wrote assembly language diagnostics for an embedded 68000 processor controller.

Designed and developed in-circuit ATE test programs for various digital and analog boards.

Designed and developed stand-alone assembly programs for the test and integration of UNIBUS device controllers.

Developed data acquisition and analysis software to drive GP-IB based spectrum analyzers. These programs greatly eased mechanical and electrical engineer's job in characterizing MEBES III and AEBEL 150 Electron Beam Lithography machines.

As the principal software designer of a two member team, developed and implemented the test strategy for the test of an Electron Beam Lithography data path. The software, written FORTRAN 77, used the host machine to drive the system under test and a logic analyzer for capturing expected output. This resulted in:

- up to an 80% reduction in test time
- a unified test approach across test departments
- the elimination of custom hardware (a one time savings of $150K).

After implementation, trained test personnel, produced extensive on-line help screens and wrote demo programs.