

David James Benn
73 Second Avenue, Klemzig, SA, 5087
08 8266 6508 [home]
0407 261163 [mobile]
dbenn@computer.org

Career Summary

A Software Developer, with more than 15 years of experience in a variety of domains.

Possesses significant experience and knowledge of all aspects of the software engineering process.

Has acted in lead technical roles on projects, mentoring less experienced engineers, liaising with customers and end-users.

Key Strengths

- General skills in design, coding, unit testing, reviews, system testing, requirements gathering and analysis, and configuration management.
- Experienced in a range of domains and software development methodologies.
- Experienced in object-oriented design, languages, and design patterns.
- Team oriented, encouraging the use of common procedures, practices, and tools.
- Test focussed, emphasising the importance of regression and repeatability.
- Exposure to a variety of programming paradigms apart from object-oriented, including functional programming.
- A strong commitment to life-long learning, enjoying teaching and mentoring.
- Innovative in problem solving.
- Excellent written and verbal communication skills.
- Obtained a security clearance in 2012.

Domains

- Science and Research Data transformation, collection, citation, and search.
- Banking protocols (e.g. BPay, direct debit, credit card gateways).
- Embedded software (smartcards, mobile phones, micro-controllers).
- Development, test, and verification tools.
- GUI development, redesign and maintenance.
- Relational databases and object-relational mapping.
- Hardware simulation and custom debugging tools.

Programming Languages and Tools

- Java (desktop and server-side) and frameworks such as Swing, Zk (AJAX), Hibernate, Spring
- C, C++ and related tool-chains (e.g. GNU) and IDEs (e.g. Visual Studio, Eclipse)
- Python, Perl, R, shell
- SQL (MySQL, Oracle, Postgres), JDBC
- XML standards and technologies
- Exposure to many other languages, e.g. Scala, Haskell, Ruby, Lisp, Tcl
- ANTLR compiler construction language
- Eclipse (for Java, C++, Python, Perl)

- JUnit, JCoverage, other xUnit variants
- Version control tools: Subversion, git, CVS, RCS, Clearcase
- Continuous Integration tools (e.g. Jenkins, Cruise Control)
- Collaboration tools such as Jira and Confluence
- Crucible for online code reviews
- Build tools: Ant, make, Scons, CMake
- ECMAScript (JavaScript), Asynchronous JavaScript and XML (AJAX)
- Brief commercial exposure to .NET (C#, Nant, NUnit, NCover) and some non-commercial .NET experience
- Test coverage, memory leak detection, lint, profiling tools for C, C++, Java
- Static analysis tools for Python (e.g. pylint), C, Java (e.g. FindBugs).
- Use of processor-specific simulators (e.g. ARM, PowerPC, PIC)
- UMLet, Borland Together and Rational Rose for object-oriented analysis and design
- VMWare, VirtualBox, VNC, RDesktop
- Microsoft Office, Open Office, FrameMaker

Professional Experience

CSIRO, October 2012 – Present time

Science Data Specialist

There are two, often overlapping, aspects to this role:

1. Working with researchers, information specialists, and developers across CSIRO to create searchable, citable data collections (and associated metadata) in the CSIRO Data Access Portal (DAP; <https://data.csiro.au>) as a member of the Research Data Service and Science Data teams.
2. Development of scripts, utilities, and plug-ins using various programming languages and the definition and documentation of workflows on high performance compute infrastructure.

Communication is an important aspect of this role. Being the only team member in Adelaide requires effective organisational skills and the ability to work unsupervised, and to make good use of email, instant messaging, and collaboration tools such as MeetingPlace, Confluence, and Jira. For code and related artefacts, revision control tools such as Subversion and git are used.

The role also involves regular and ad hoc data collection administration and reporting tasks.

Achievements

- Developed code in a variety of programming languages (Java, Python, R, C++) for science data processing projects including:
 - A C++ plug-in for a workflow tool (Workspace) to read 3D mesh files from a commercial computational fluid dynamics tool, and
 - Scientific Python code to transform NetCDF oceanographic data.
- Assisted researchers in the creation of research data collections.
- Communicated with staff in various groups across CSIRO such as oceanographic and atmospheric research, radio astronomy, plant phenomics, computational informatics, electrical technologies, national collections. Helped to define data workflows including co-authoring an international workshop paper with the plant phenomics group regarding management of high resolution plant data.
- Presented at Computational and Simulation Sciences conference (2014) on work towards a HPC pulsar processing pipeline and integration of DAP with a workflow tool (Workspace).
- Co-authored a poster paper about aforementioned HPC pulsar processing pipeline for eResearch Australasia 2014.

- Presented on DAP at Barr-Smith library, University of Adelaide for an eResearch SA event in 2014.
- Exposure to the Map-Reduce framework Hadoop, making tools more accessible, providing examples, contributing to discussions about the place of Hadoop alongside existing distributed computing approaches.
- Presented at two Big Data workshops (2012, 2013) and was on the programme committee for the second.
- Performed a variety of ad hoc data collection administration operations in communication with DAP developers and Research Data Service colleagues.
- Carried out regular (weekly and monthly) and ad hoc DAP report creation.
- Worked with information specialists and developers to identify bugs and define future requirements for the DAP.
- Encouraged use of the collaboration tool Jira within the Science Data team.
- Participated in an employee return-to-work programme as a mentor in conjunction with the CSIRO Computational Informatics group.
- Supervised a summer vacation student.

CSIRO, February 2012 – October 2012

Java Software Developer

The focus of this role was development of the CSIRO Data Access Portal (<https://data.csiro.au>) in a well-oiled team of programmers, testers, business analysts, and information specialists.

Programming languages, frameworks and tools used were: Java, JavaScript, Zk (AJAX), Hibernate, Spring, SQL (Oracle and Oracle Developer), Eclipse, Maven, Subversion, Jira, Confluence, Jenkins.

Achievements

- Development of functionality for the CSIRO Data Access Portal (<https://data.csiro.au>) over multiple iterations (sprints) using the Agile methodologies Scrum and Kanban.
- Created unit tests, worked within a continuous integration environment (Jenkins), wrote documentation, demonstrated new functionality to stakeholders.
- Attended daily stand-up meetings to discuss progress, future work, and blockers.
- Took on the role of Scrum Master, as was expected of all team members over time.
- Took part in on-line code reviews (with Crucible) and security-oriented code inspections.
- Made daily use of collaboration tools such as Confluence and Jira.
- Engaged in build and release activities in addition to planning, estimation, and end of sprint reviews.

Australian Semiconductor Technology Company, Adelaide Jun 2007 – February 2012

Software Developer

Development and test of simulation components and tools for Systems on a Chip (SoC) in C, C++ and Python.

Achievements

- Development of multiple SoC component models and C driver code.
- Bare-metal ARM simulator development and testing.
- Debugger integration for simulations with ARM, PowerPC, and other cores.
- Change control role for internal tools and libraries.
- Introduced static analysis tools into regression test environments.

Internode, Adelaide

May 2006 – Jun 2007

Analyst Programmer

Design, code, and test of internal Java and Perl software systems, primarily payment processing

related.

Achievements

- Development, unit, and integration testing with Java 5, Eclipse, JUnit, Maven, Cobertura, Hibernate, Tapestry, Spring, Perl, Postgres, Cruise Control.
- Development of banking protocol systems: BPay, Credit Card gateway access, Direct Debit.

Worldsmart Technology, Adelaide

Apr 2006

Senior Software Engineer

Custom WinCE device driver and C# (.NET 2.0) development for Adelaide hotel industry.

Achievements

- Introduced a test-driven approach and tool flow (Nant, NUnit, NCover, TestDriven.NET, suggested CruiseControl.NET).
- Played a leading role in finding and fixing a non-deterministic device driver bug that had eluded others.

Freescale Australia SoC Technology Centre, Adelaide

Jul 2004 – Mar 2006

Senior Staff Software Engineer

Motorola's Semiconductor Products Sector was spun off as a separate company in Dec 2004, initially as a subsidiary of Motorola in April 2004, with the name of Freescale. Freescale designs and manufactures a variety of integrated circuits (ICs) and Systems on a Chip such as 68000 and PowerPC derivatives.

Achievements

- Worked in a US-Australian team to develop an object-relational mapping layer to replace an existing, costly, and undocumented layer, as part of an IC design verification tool. Used Java 2, JDBC, MySQL, CVS, Eclipse, Ant, Groovy. Applied Gang of Four design patterns, and data access patterns.
- Worked with a colleague to develop a regression test suite as part of this work, and with the rest of the team to improve collaborative development and software quality practices (primarily testing and change control). Used JUnit, JCoverage.
- Spent two weeks visiting and working with the rest of the team in Austin, Texas.
- Provided support and training to Adelaide project team members using this IC design verification tool.
- Improved the software development capability of Freescale Australia by helping to organise and presenting at technical discussions, writing application notes, and encouraging cross-project knowledge sharing.

Motorola Australia Software Centre, Adelaide

Apr 1997 – Jun 2004

Software Engineer, Senior Staff Engineer

Achievements

- Developed part of a formal verification tool for embedded system model-based designs on a 10 person team. C++ and STL were used along with ANTLR for compiler development. Used the Borland Together tool for design and limited round-trip engineering in this and other projects at Motorola.
- Redesigned the GUI for this formal verification product so that it could be used with three variants of the tool. Used Java and the Swing library for GUI development. The efficiency and maintainability of the GUI code base were also improved.
- Worked with 3 others to develop an industrial strength SAT solver for use in design verification (primarily hardware) by studying the literature. C++ and STL were used.

Understanding the computational complexity of proposed implementations was particularly important in order to maximise speed and minimise memory usage. Used Perl and Python for testing, debugging and other utilities. Used Gang of Four design patterns here and in other 2 projects above.

- Worked on a 25 person smartcard operating system project as a technical lead and developer. An embedded systems project with extreme resource limitations and security requirements. C was the main language, with some M*CORE (32-bit RISC) assembly used for time-critical code.
- With a team of 6, prototyped a JavaCard smartcard development and testing environment, then as technical lead of a smaller group created a JavaCard virtual machine for a product based upon this prototype. These were demonstrated at trade shows and tested by potential customers. In the latter stages, took on the role of Software Architect to coordinate three technical leads and plan for the product's future with an eventual team of 25 developers.
- Worked with 2 other developers to prototype the modification of a Motorola mobile phone operating system in order to interface with a barcode reader. C was the development language.
- Worked alone to create a smartcard reader protocol stack in C (Visual Studio 6).
- Took on a project lead role for a team of 7 software engineers to develop a WAP-based system. Project plan development, customer liaison, reporting, and scheduling were the primary responsibilities.

University of South Australia

2001 – 2005

Sessional tutor and guest lecturer

Wanted to maintain contact with the University to share industry experience with students, and maintain contact with University research and teaching trends.

Achievements

- Delivered reverse engineering lecture, Information Systems Maintenance and Re-engineering subject (2004, 2005, 2006).
- Tutored for a first year Programming in Java course (2005).
- Lectured and tutored in Intelligent Systems Technology (Lisp; 2001, 2003, 2004).

Vision Internet

Dec 1995 – Apr 1997

Unix System Administrator and Programmer

- Vision Internet's system administrator:
 - Responsible for the day-to-day running of the BSDi and FreeBSD Unix Systems of Vision Internet Services;
 - Ensured the integrity, availability, and confidentiality of the data in my care. Vision had in excess of 1000 users by around mid-1996. Other duties included installation of a variety of Unix programs as needed, applying security patches, providing user support, liaising with Federal police and AUSCERT during a security breach.
 - Vision Internet's programmer, working primarily with C, Perl, and shell scripts. Wrote custom CGIs for use with web pages, and miscellaneous system utilities as required.
 - Primary software development was an online accounting system which worked with the raw Unix and Cisco terminal server accounting logs, handled monthly invoicing, daily time debiting, session termination, and general time-usage administration and notification. This was an ongoing project since Vision's account types were fluid, and the company's state-wide presence continued to grow. This software was also sold in 1996 to another Tasmanian Internet provider, and custom updates were provided as necessary.

University of Tasmania**Feb 1993 – Nov 1995*****Level A Academic in Applied Computing***

- Taught a variety of undergraduate computing units. Developed course structure and teaching material for several of these units.
- Mostly small classroom and computer laboratory situations, but also developed and delivered lectures for computer security, programming paradigms units.
- Assessed work carried out by students during the semester and in formal examinations. Designed numerous assignments and in a few cases, final examinations. Second assessor for other units' exams.
- Participated in departmental research groups including Multimedia and Expert Systems, although teaching was the main focus.
- Contract was not renewed due to funding cuts, despite a recommendation by the head of department for another 3 year contract.
- Taught the following computing subjects during this period:
 - Programming and Problem Solving with Pascal
 - Data Structures and Algorithms
 - Computer Security
 - Introduction to Operating Systems, Principles of Operating Systems
 - Expert Systems
 - Programming Paradigms
 - Computer Concepts (transistors to processors to assembly language to operating systems)
 - Computing Systems (word processing, spreadsheets, databases)
 - Computer Networks and Digital Communications

University of Tasmania**Feb 1992 – Dec 1992*****Part-time tutor in Applied Computing***

- Enjoyed the minor teaching component of CSO position below, so chose to take up a tutoring position.

University of Tasmania**Jul 1991 – Feb 1992*****Computer Systems Officer***

- Computer Systems Officer (CSO) for the Technical Services section of the Department of Applied Computing with wide-ranging duties: Macintosh and PC software support, hardware (including network) installation and maintenance, Novell Network administration, Unix and VAX/VMS system administration, Unix, Mac, and PC programming (C and Pascal), acting as a resource person on technical matters for students and staff, running short courses on various aspects of Unix.

University of Tasmania**Feb 1991 – Jun 1991*****Trainee Computer Systems Officer***

- Worked with a variety of computing systems (Mac, PC, Unix, VAX/VMS), wrote programs for teaching purposes, and laid the groundwork for becoming a jack-of-all-trades CSO.

Registered Nurse, Launceston General Hospital**Feb 1987 – Jan 1990**

- Intensive/Coronary Care, Operating Room, and Recovery Room.

Registered Nurse, Kimba District Hospital**Dec 1986 – Jan 1987**

- Geriatrics, Casualty, Outpatients.

Various employers through Adelaide agencies**Feb 1985 – Jun 1986**

- Geriatrics, Private Medical/Surgical, High Dependency, Intensive Care.

Registered Nurse, Queen Elizabeth Hospital**Feb 1984 – Feb 1985**

- Urology, Ear-Nose-Throat, Neurology, Neurosurgery, Ophthalmology, Rheumatology, Recovery.

General Nurse Training, Queen Elizabeth Hospital**Feb 81 – Dec 1983**

- Gained the experience and qualifications necessary to practice as a Registered General Nurse.

Qualifications**Master of Science (Computer and Information Science)****2001**

University of South Australia.

Bachelor of Applied Computing**1992**

University of Tasmania.

General Nursing Certificate**1984**

Queen Elizabeth Hospital.

Professional Memberships and Affiliations

- Member of the Australian Computer Society (ACS).
- Affiliate member of IEEE Computer Society.
- Member of Association for Computing Machinery.

Ongoing Development

- Attended and presented at Big Data workshops (Sydney) in 2012 and 2013.
- Attended Supercomputing in R workshop in 2012.
- Attended eResearch 2012 and 2013 conferences.
- Attended Computational and Simulation Sciences conference in 2012.
-
- Joint editor of Second Conceptual Structures Interoperability Workshop (2007).
- AI 2006 and 2007 reviewer.
-
- Presented papers at the International Conference on Conceptual Structures 2001 at Stanford University (<http://www.users.on.net/~dbenn/Masters/>). Accepted invitations to be on the Program Committee for ICCS 2002 and 2003.
-
- Frequent attendance at local branch ACS, PIC, Java, and .NET group meetings.
- Training between 2003 and 2006:
 - ACM Introduction to SQL;
 - Introduction to Microsoft .NET development (2 days, Kaz Training);
 - RTL hardware design and verification with Verilog (2 days, in-house);
 - MySQL training (1 week certification level course, MySQL in Austin, Texas);
 - Verilog assertion-based functional verification training (1 day, in-house);
 - Functional verification of Verilog using OpenVera (3 days, in-house);
 - DotNetNuke training (1 day, Kaz Training, Australian .NET User Group);
 - OO design, design patterns, advanced C++ (in-house, 3 half days);
 - TI MSP430 day seminar (July 2006).
 - Other training from 1997 to 2002: Structured Analysis and Design, RTM (requirements

traceability and management tool), Tcl/Tk scripting, Project management, Estimation techniques, Capability Maturity Model, External Rational seminars on Configuration Management, OOAD, RUP, Embedded Design Automation domain training, Object-Oriented Analysis and Design with UML, Requirements Analysis with Use Cases, Standard Template Library, Extreme Programming, Time management.

Awards

- Recipient of the 2011 Director's Award from the American Association of Variable Star Observers for leading the development of VStar. See <http://www.aavso.org/directors-award>
- Recipient of The Bill Bradfield Astronomy Award in 2012 from the Astronomical Society of South Australia for the development of VStar. See <https://www.assa.org.au/membership/awards>
-
- Support Achievement Award in 2013 from CSIRO Mathematics, Informatics and Statistics (became CSIRO Computational Informatics then Digital Productivity).

Personal Projects

- Since 2009, a variable star data visualisation and analysis tool called VStar. See <http://www.aavso.org/vstar-overview> for details.
- C library of PIC micro-controller driver functions for LCDs, keypads, Dallas Semiconductor 1-wire protocol, 7-segment display multiplexing, state machines, timers, switch input, and so on. The code has been written in such a way as to be easily ported to other micro-controllers such as AVR. Various applications of the library, e.g. dimming night light using pulse width modulation, temperature logger (<http://sourceforge.net/projects/picclib>).
- Free compilers and interpreters
 - ACE Basic for the Amiga. Generates 68000 assembly (1991–1996). Favourably reviewed in several magazines. ACE had a very active community, and the author remains amazed that people are still using it either on Amigas or Amiga emulators.
 - LittleLisp for the Newton PDA (2000), a Lisp system written in NewtonScript.

Other Activities and Interests

- Amateur astronomy. Member of the Astronomical Society of South Australia, American Association of Variable Star Observers, and Variable Stars South.
- Bike riding, walking, playing golf.
- Reading science, philosophy, history, science fiction.
- Generally spending time with wife Karen, son Nicholas (13), and daughter Heather (10).

Referees

- Gareth Williams, Gareth.Williams@csiro.au, 03 8601 3804
- James Dempsey, James.Dempsey@csiro.au, 02 6214 2912

Others available upon request.