

Robert Hook

17 BARGO ST ARANA HILLS, QLD 4054
TEL 04 6650 6633

rahook@gmail.com
<http://www.parttimepolymath.net>

Profile

An information technology professional with over 24 years experience in the industry, Robert Hook has been involved in a wide variety of projects that have required and fostered a broad and comprehensive skill set.

Robert's main skills revolve around the design, construction and maintenance of large information systems based around relational databases, primarily in a UNIX environment, along with the design, construction and maintenance of web-based information systems, frequently backed by a database. Many of the systems he has been involved with have been based around Ingres, Oracle and Postgres and he has considerable experience as a database designer. In recent years he has been largely occupied with the design and construction of sophisticated Web Services and Web Applications written in Java using J2EE1.4 and J2EE1.5 paradigms and approaches.

He is a highly competent analyst and programmer, with a dedication to producing high quality, reliable and efficient solutions in a timely and cost-effective manner. Robert has a noted ability to rapidly assimilate new technologies, and actively monitors relevant industry best practices and standards in order to continually improve his abilities to deliver solutions.

Experience

SOFTWARE ENGINEER, TRANSACTION NETWORK SERVICES – MAY 2010 - PRESENT

Working in the Continuous Engineering team to provide third level engineering support to a range of web-oriented Java-based products for the payment processing industry. The role requires finely honed problem solving and resolution skills, and the ability to maintain a balance between the needs of the customer and the needs of the enterprise.

SOFTWARE ENGINEER, SALMAT/HPA – 2003 - PRESENT

Responsible for the design and development of various internal software products, particularly oriented around Java-based web services. The past three years have been devoted to fast, high-availability complex web applications to support the NAPLAN initiative.

DATABASE ADMINISTRATOR / PROGRAMMER, QLD POLICE SERVICE – 1998-2003

Information Systems Branch. Database administration. Responsible for planning and implement upgrade of Ingres 2.0 to Ingres II across the enterprise. Database design and analysis. Design and implementation of large scale data conversion and cleansing strategies, processes and tools. Design and implementation of Database Administration tools and processes.

DATABASE ADMINISTRATOR / PROGRAMMER, QLD DEPARTMENT OF NATURAL RESOURCES – 1995-1998

IVAS, IVASe, LGIP projects. Database design and analysis. Design, implementation and maintenance of IVAS system maintenance tool set. Tool and system design and implementation for data acquisition phase of LGIP project.

SENIOR ANALYST/PROGRAMMER, DATABASE ADMINISTRATOR, PINE RIVERS SHIRE COUNCIL – 1989-1995

Responsible for design and maintenance of broad range of administration and financial systems.

PROGRAMMER / TECHNICAL SUPPORT, SHANNON ROBERTSON SYSTEMS – 1988-1989

Development and support of PC based small business systems and support systems for the agricultural industry.

SECONDARY SCHOOL TEACHER, MATHEMATICS AND SCIENCE, QLD DEPARTMENT OF EDUCATION – 1987

Robert remains a registered teacher and maintains his professional interest in educational and didactic techniques, policies and trends.

Education

Griffith University – Bachelor of Science, 1986

Queensland University of Technology – Graduate Diploma of Teaching, 1986

Mitchelton State High School – Senior Certificate, 1982

Skills

Highly Competent

Java, Ant, JUnit, JSF, Struts, JBoss, Apache Tomcat, NetBeans, JBuilder, JSP, J2EE Web Services, J2EE Web Applications, C, C++, Ingres ABF, XML, XHTML & HTML, CSS, Oracle, Postgres, SQL, JDBC, Unix, Systems Analysis and Design, Data Analysis, Database Analysis and Design, Ingres Database Administration, Technical Writing, Team Leadership, Systems Maintenance, Systems Programming, Source and Document control processes, Product Release processes, Quality Assurance systems, Training, and Mentoring, Web design and construction.

Competent

Maven, Eclipse, IntelliJ, XSL/XSLT, UML, Design Patterns, JavaScript, MySQL administration, Cryptography and Security standards and protocols, MUMPS, Basic, Fortran, Pascal, AppleScript, dBase III+/Clipper, CGI Scripting, Unix Systems Administration, Project Management, Image preparation for Web presentation.

Familiar With

AJAX, iOS Programming, Objective-C, Forth, Assembly language programming, Linux Systems Administration, Carbon and Cocoa programming for Mac OS X.

Technical Interests and Aspirations

Robert is greatly interested in the issues associated with the management of large and complex information systems, particularly data integrity, security and privacy. He is also particularly interested in user interface design and in adopting new technologies to present legacy information stores.

He greatly enjoys working with various Java-based techniques for delivering interactive content and services through the Web, leveraging the power of open-source and open-standards products and XML solutions.

Transaction Network Services, May 2010 - Present

Currently Robert is working within the Continuous Engineering team providing third level support for a variety of cutting edge and legacy products for the Payment Industry. The role requires sophisticated and rapid problem solving, prioritisation and resolution skills, and a pragmatic approach to providing solutions that satisfy both the end customer and the enterprise. A significant component of his work in this role is overview and maintenance of software standards for new products, and continuous improvement of the security, reliability and maintainability of legacy systems.

The products in place are all internet facing and oriented around secure communication of transaction data, backed up by strict adherence to and compliance with PCI-DSS. All products are written in Java leveraging the powers of Spring for resource injection and a variety of other modern technologies including JPA and JAAS. The development environment is Agile, with a very strong emphasis on automated unit, integration and regression testing coupled with a traditional staged release environment.

Both Eclipse and IntelliJ were used for development, with builds brokered by a mixture of Ant and Maven. All development and maintenance activity was performed against a Subversion repository, and deployed onto system and integration test hosts via a continuous integration environment based around Hudson. In house documentation was written and published via a Confluence CMS instance, and Robert was a significant and avid contributor to this documentation.

During his time at TNSi, Robert participated in PCI-DSS mandated security training, and is well abreast of current issues and solutions related to web-facing systems, and in particular to security, confidentiality and auditing of financial systems. Some of his successes in his role at TNSi include significantly increasing and improving internal documentation, implementing a controlled and documented production mirror for the CE team, and improving the performance of a financial batch management system by an order of magnitude.

HPA/Salmat, October 2003 - May 2010

Robert initially began at HPA working within the Queensland branch IT group, but was promoted in February 2004 to the national R&D group. In November 2007 HPA became part of Salmat, but Robert's role remained the same. During his time within HPA, he has been responsible for or heavily involved with the design and creation of a number of key applications critical to the success of the company. These all related to the core business of the company, which is to say with capturing, transforming and printing very large volumes of data. The bulk of the work done involved the creation of a set of inter-related Web Services, all written in Java.

This activity was part of the expression of a corporate IT strategy wherein a suite of services, each fulfilling some core business activity, or providing a core facility, would be loosely coupled together and spread across the corporate network as required. This strategy saw these services deployed as Java-based web services running within a full J2EE environment using a number of different application servers, including JBoss, Orion and Tomcat. Client side applications were written in C# using the .Net framework, while server-side relied heavily on Axis and related technologies. More recent versions of these services migrated to using JAX-WS. Robert was one of two developers solely responsible for the design and implementation of all server side components.

The last few years were devoted to the development of a sophisticated solution to support on-line marking for the NAPLAN initiative. This solution involved presentation of images through a Flex interface provided by a pair of sophisticated web services and web applications. It was characterised by its emphasis on high availability and high performance, driven by very aggressive response requirements under high load. Robert championed the introduction of Continuous Integration (CI) methodologies during this project, and oversaw the installation and configuration of a build environment based on Hudson to support this approach.

As well as the design and implementation of these systems, Robert was responsible for research and implementation of industry best practices in the J2EE and Web Services realm. This led to the adoption of a single application server – JBoss 4.x – and the adoption of a wide range of industry standard tools and practices. By adopting these open source solutions and standards, and formalising the approaches taken to the design and implementation of J2EE solutions, significant cost and time savings were realised.

As the Web Services were being deployed within a full J2EE compliant application server, they were enhanced by the addition of web-based administration and configuration interfaces written using Struts, and in later iterations JSF, and contained and delivered their own documentation. Each application was also accompanied by one or more client libraries used to facilitate the construction of client-side applications, and these libraries were distributed within the application deployment package. All web services were initially constructed using various versions of Jbuilder, then migrated to NetBeans, as these tools provided powerful mechanisms for simplifying construction of J2EE products.

Several principles guided all the development performed by Robert during his time with HPA, and these were a combination of his personal vision and the corporate vision. As far as possible, all the server-side products were designed to be as close to zero-configuration as possible, with the intention being that all support staff would need to do was deploy the service, then continue configuration through web pages, without any of the typical manipulation of configuration files prior to deployment. Robert also championed the philosophy that documentation and training are core deliverables, and that no product could be considered complete without providing these. Taking that a step further, in collaboration with other developers he was instrumental in ensuring that traditional testing, quality control and release management procedures were put in place and maintained.

His dedication to quality and robustness, and a marked willingness to work whatever hours were necessary to fulfil corporate objectives and requirements, saw Robert lauded on several occasions through the national Employee Recognition program.

Queensland Police Service, December 1998 - August 2003

Robert's time with the Information Systems Branch (ISB) of the Queensland Police Service was spread across three primary streams of activity: Senior Analyst / Programmer for data conversion and cleansing projects, Database Administrator, and technical lead and mentoring.

He was initially bought into ISB to design, construct and implement a solution for the take-on of Offender History data into the Polaris system from a legacy system. This project required the development of a suite of tools that would be able to process a very large data set as quickly and efficiently as possible, while performing sophisticated data conversion and cleansing. As well as dealing with the difficulties of merging several disparate and complex data sets, the extreme sensitivity of the data required that the tool set produced accurate and unambiguous audit and reconciliation data. The detailed implementation plan developed by Robert for this project was successfully applied in early 2000.

It became evident in the initial stages of this project that ISB had a need for a dedicated data conversion and cleansing arena, and for a well-designed and documented set of reusable tools, solutions and software components. Robert took the lead in the specification, design, construction and implementation of this data conversion and cleansing (DCC) environment. This included reusable component libraries and fundamental tools, in addition to programming, design and documentation guidelines and standards.

The DCC arena proved of great benefits for subsequent projects. The Microfilm conversion process and tool set was designed by Robert and implemented in late 2000 by junior programmers under his guidance. Two other major projects in 2001 also reused the environment.

Through 2001, Robert concentrated on Database Administration activities. During this period he was responsible for the design and implementation of a corporate wide rollout of Ingres II 2.0 to replace older versions of the product. Initial stages of the project required comprehensive regression testing of the new product as well as the development of standards for installation, upgrade and configuration. In addition to detailed project planning, considerable time was spent mentoring junior Database Administrators, significantly upgrading the Ingres operations and other documentation, development of programming standards, and rewriting the local Ingres database synchronisation product.

He returned to DCC activities in 2002 with the Missing Persons project. This project was not restricted to code design and construction, but was explicitly used to investigate more effective ways of delivering solutions. The suite of reusable components was overhauled and replaced with a comprehensive set of C++ classes based on the Relational Database Access Layer Pattern Language, with various supporting and utility classes to encapsulate other data sources and sinks for DCC products. Deliberate investigation of reusable Open Source solutions and code fragments was performed, and such code adopted where appropriate and possible. Product configuration was redesigned to use XML files, and intensive use was made of inline documentation and automatic documentation generators.

The design and construction methodologies championed by the Agile Alliance, particularly elements of the Extreme Programming (XP) and Pragmatic Programming approaches, were investigated and applied to this project wherever possible, and informed a revision of the DCC programming standards. During this time, Robert was able to apply these new ideas in the creation of the Polaris Code Review Process and Polaris Program Design and Specification standards.

During his time at ISB, Robert's role was frequently to act as technical lead and mentor in both the Database Administration and Analyst/Programmer realms. As well as significant amounts of time dedicated to physical database design and implementation, to data analysis, and to technical support to other Programmers and Database Administrators, Robert was often responsible for finding solutions to the more difficult and complex problems. Beyond simple resolution of such issues, he used them as opportunities to improve documentation and tools, perform ad-hoc training, and to encourage reformation of organisation wide policies and procedures.

Some of the smaller projects were particularly satisfying to Robert. These included the Polaris PES data take-on which successfully demonstrated the cost benefit of the reusable foundation and new methodologies introduced for Missing Persons, the revision of the NNI download which allowed QPS to increase the details provided to this national database from Polaris almost tenfold, and the Batch Recommend Link tool which could find all Persons apparently stored more than once in Polaris within twenty minutes by traversing more than 20 Gigabytes of data.

Robert's final months with the Queensland Police Service were divided between resolving long-outstanding data integrity and data quality issues, and advising on design and construction for the DNA DCC project. This included ad-hoc training of junior programmers in C++ and use of the revised DCC framework.

Queensland Department of Natural Resources, October 1995 - December 1998

Robert was associated with three projects while working for DNR. These were all related to the property valuations business area, namely IVAS, LGIP and IVASe. This allowed him to draw on his experiences gained from six years working in Local Government.

When he came to IVAS, Robert was acting as Database Administrator for what was, at the time, one of the largest Ingres databases in the world. Beside the day-to-day administration activities, he undertook a substantial revision of the tools and processes used for administration, massively automating a great part of the standard administration and monitoring activity. While there, he managed to reduce the number of required full time database administrators from two to less than one. He also brought substantial performance and reliability improvements to the IVAS database through changes to the physical database design and implementation. Part of the data quality improvements were delivered through a tool which performed comprehensive relational integrity checks across ten independent physical databases and was able to find any and all integrity errors within a few hours.

As part of the revision and automation activities, Robert created a comprehensive set of documentation for the site, which he eventually turned into a Web-based documentation repository hosted on a Linux system. This intranet server also hosted a simple interface to the database monitoring suite, allowing the business representatives to gain a quick overview of the system status directly. Toward the end of his time with IVAS, Robert further extended this Linux system to provide a means for the business representatives to export data to magnetic tape on an ad-hoc basis via a simple Web interface without the intervention or assistance of technical support staff.

Beside Database Administration and physical database design, Robert was responsible for technical support to the team of C and Ingres/ABF Analyst/Programmers who maintained the IVAS application. This not only included code review and assistance with program design, it involved the creation of various coding standards and guidelines.

The LGIP project entailed a large-scale cleansing of name and address data in IVAS. Data was obtained in a wide variety of formats from every Local Government Authority in Queensland, parsed into a form where it could be compared to IVAS, then used to correct IVAS as required. Robert was responsible for the design, construction and implementation of all code for this project. The main tools were the parsers and data converters running under the UNIX environment, and a VisualBasic data entry tool used for manual corrections prior to loading to IVAS. As well as their ability to process very large quantities of data quickly and efficiently, the LGIP tools were distinguished by their sophisticated algorithms for parsing names and addresses out of more-or-less arbitrary input data sets.

Throughout his time at DNR, a parallel project was progressing to replace IVAS with a client-server solution using Ingres databases, Tuxedo as a transport layer and VisualBasic for presentation and data entry. For this enhancement project, IVASe, Robert contributed the majority of the physical database design including specification of new server infrastructure. The move to the use of a PC-based application development environment allowed Robert to create a database schema versioning system which largely automated the implementation and maintenance of the physical database schema for IVASe. This system remains in use and has been recognised as a good general solution to the problem of bringing version control to the maintenance of an Ingres database schema.

Pine Rivers Shire Council, June 1989 - October 1995

The six years spent at PRSC involved Robert in the maintenance, design and development of a very broad range of financial and administration systems to handle everything from dog registrations to the library catalogue. Initially these systems were developed using Digital Standard MUMPS and RSTS/BASIC on PDP-11 minicomputers. Over time council adopted PCs and moved to a UNIX base for several Ingres databases and a GIS system. This entailed a move to programming in C, ABF and Ingres/Vision.

Working in a small multidisciplinary team gave Robert exposure to all aspects of computer systems maintenance and development, and even gave him the opportunity to manage the IT section for almost six months in 1995. The extremely varied challenges posed at this site fostered the development of a strong belief in the importance of a flexible and responsive approach to providing fast, cost-effective solutions.

Shannon Robertson Systems, January 1988 - April 1989

Robert began his career in information technology with Shannon Robertson Systems, trading as Saltbush Systems, a small company which at the time held a dominant position in the agricultural software market. He was responsible for the creation and maintenance of farm management and small business software to run on MS-DOS PCs based around various database products including dBase III+/Clipper. As well as programming, Robert performed client support and training, PC hardware maintenance, and was even called on from time to time to help in dispatch and shipping.

Robert's first exposure to UNIX occurred at Shannon Robertson, where he was responsible for the design, construction and implementation of a Xenix-based office automation solution distributed across the offices of a legal firm scattered over the Sunshine Coast.

Queensland Department of Education, January 1987 - December 1988

Having developed an interest in Computer Aided Education (CAE) and Computer Based Training (CBT) while studying for his undergraduate qualification, Robert accepted an offer to study teaching. After gaining that qualification, he opted to put it to use for a year, and spent that year as a secondary school teacher of Junior mathematics and science in a small country town. While greatly enjoying teaching and working with young people, Robert preferred to return to his true calling and move into the information technology industry. Despite this, Robert remains a registered teacher, and make an effort to keep abreast of modern didactic methodologies. This has served him well as it has allowed him to effectively take on the role of technical trainer and mentor to less experienced staff.

Early Experiences, pre 1987

Robert was fortunate to be exposed to serious small computers and systems as a high school and tertiary student. This not only gave him the opportunity to understand computer technology from a very fundamental level, it gives him a perspective on the astonishing development of the industry over the last twenty years. In 1980 his high school purchased a small PDP-11 system, allowing him to spend many hours writing code in assembler, BASIC and FORTRAN, balanced by assisting his brother in the development of process control hardware by writing and debugging software in assembler and Forth.

His tertiary experiences mainly involved Pascal on mainframe and minicomputer systems, and quite a lot of instrumentation construction and programming in assembler. His third-year project saw him partnered with another fledgling programmer to write a general-purpose 3D graphing program for the recently released Macintosh, as an exploration to discover if the Macintosh could be considered a real computer. He decided it was, and has been programming personal projects on the Macintosh ever since.

Personal Interests and Background

On my business cards and personal web site (<http://www.parttimepolymath.net>), I describe myself as a "Part Time Polymath". This is mainly word-play, but it was one way to indicate that my interests are wide and varied. I am a voracious and omnivorous reader, and that pattern extends through my other interests in fine arts and music.

I am a keen amateur musician, playing renaissance and medieval recorders, and saxophone. I've dabbled with painting and fine arts, although most of my artistic endeavours now revolve around digital images. Although I am an avid technophile, particularly regarding Macintoshes, most of my leisure time is taken up by my deep interests in things medieval. I am heavily involved in 15th Century Living History and am seriously studying renaissance fencing. My historical re-enactment activities have led me to try my hand at everything from sword fighting in armour to hand sewing and shoe making. I am a past president of the Queensland branch of the Richard IIIrd Society, and was treasurer of the Queensland Living History Federation for several years. Over three years running I was significantly involved in the organisation of the annual Abbey Medieval Festival on behalf of the Abbey Museum of Art and Archaeology. When not busy with all of these things, I can be found in my workshop, working with wood.

My particular technical interests focus on two domains: the issues with developing information repositories that should be accessible from anywhere for an indefinite time into the future; and the efforts to make software construction a professional, cost-effective and efficient activity.

I have a great interest in the work being championed by the World Wide Web Consortium in various applications of XML to the issues of resource description and classification. I am firmly of the belief that applications of XML provide the possibility of constructing information repositories that are not dependent on maintenance of legacy systems into the indefinite future for access. The Semantic Web and Web Ontology projects may eventually fulfil the long-held dream of automating the conversion of data into knowledge. I would relish the opportunity to deploy some of the work being done in these areas in a real environment, but meanwhile play with the presentation aspects through my personal website.

Having spent 23 years in the IT industry, I have seen a many trends in software construction come and go, and have a fair degree of scepticism about most buzzword marketing. I am however very much drawn to the methodologies espoused by the Agile Alliance members, particularly the ideas expressed in the Pragmatic Programming writing. While I have some concern that these methodologies may be dependent on having access to experienced programmers and designers, I am convinced by my own experiences and the documented case studies of the cost benefit of these light-weight methodologies. The test-driven approach to software construction espoused in the Extreme Programming approach, coupled with the older ideas of Design-By-Contract and Literate Programming, demonstrably improve code quality and reliability. Wherever possible, I hope to introduce and adopt elements of these paradigms in an attempt to code for the future.

Skill and Experience Details

I have worked for over 23 years in the Information Technology (IT) industry, in a broad variety of roles. My main skills centre around the design, construction, implementation and maintenance of large, complex information systems. The bulk of these systems have been based on relational database systems, particularly Ingres, in a Unix environment. My abilities span four domains:

1. Analyst/Programmer, primarily with Java (particularly J2EE related technologies), XML, C, C++, Ingres ABF, and various standard Unix scripting languages;
2. System Design and Implementation, generally oriented to database-centric information systems involving Oracle and Ingres;
3. Data and Database Analyst and Designer, generally focused on relational databases and data models, but with a strong interest in alternative and evolving options;
4. Database Administrator, primarily involving Ingres.

I believe myself to be a highly competent IT professional, with a dedication to producing high quality, reliable and efficient solutions in a timely and cost-effective manner. I possess a noted ability to rapidly assimilate new technologies, and actively monitor the evolution of relevant industry best practices and standards in order to continually improve my ability to deliver appropriate Information Systems (IS) solutions.

Development

My career as an Analyst and Programmer has involved me with the creation, implementation and support of a very broad range of Information System solutions for a variety of organisations:

1. Hermes Precisa Australia/Salmat, R&D Branch, October 2003 – present;
2. Queensland Police Service (QPS), Information Systems Branch, December 1998 – August 2003;
3. Queensland Department of Natural Resources (DNR), IVAS and IVASe projects, October 1995 – December 1998;
4. Pine Rivers Shire Council (PRSC), June 1989 – October 1995;
5. Shannon Robertson Systems, January 1988 – April 1989.

Prior to my first professional engagement with Shannon Robertson Systems, I also had a good deal of informal experience as a programmer both through my tertiary studies and High School, including informally assisting in the development of firmware for process control systems for James Hardy Industries in the early 1980s.

I possess very strong programming skills with:

- Java (particularly J2EE technologies)
- XML
- C
- C++
- Unix Bourne/Korn shell and related Unix tools
- SQL
- XHTML and HTML
- CSS
- UML

I am also comfortable using and competent with:

- Ingres ABF, including Vision
- MUMPS
- various implementations of Basic
- JavaScript / ECMAScript
- XSLT
- MS-DOS batch scripts
- AppleScript
- Fortran
- Pascal
- dBase III+/Clipper

I am familiar and have had some experience with

- Objective-C
- Forth
- Assembly language programming for Motorola 6502 and 6800 series microprocessors, and Intel 8086/8088 series microprocessors.

As an Analyst/Programmer and technical lead over my career, I have been responsible for and involved with all stages of software development from requirements gathering and definition, through design and specification to coding and testing. I have worked within a wide variety of development paradigms including XP, Agile and traditional waterfall models, and am entirely familiar with SDLC.

Development for Transaction Network Services

The Continuing Engineering team is responsible for third-tier customer support engineering, maintenance of deployed products, and continuous improvement of the entire code base. As such the majority of development work is intentionally small in scope and focussed heavily on getting the right solution out into production as quickly as possible, right the first time, and right every time. A blend of Agile and XP methodologies are used within the team, coupled with an absolute adherence to rigorous unit, integration and system testing. The deployed system is extremely complex, with three separate deployment stacks around the world, and multiple loosely coupled services on multiple hosts within the stack.

Development for HPA/Salmat

My responsibility at HPA/Salmat was to construct and design a suite of complex web-based applications and solutions, primarily within a J2EE environment. As this role was within a national R&D group, my responsibilities also included rigorous research and appropriate adoption of industry best practices and emerging technologies. This led to me taking a lead role in adopting open source solutions and more rigorous and formal development practices, and in evangelising these approaches through state branch IT groups.

Coding within HPA R&D involved the use J2EE Java (1.4.2 and 1.5) on the server running under a variety of Linux and Windows versions. Most coding was carried out using various versions of Jbuilder and NetBeans, and considerable use was made of open-source components, particularly those from the Jakarta Commons and related projects.

Industry standard best practices from the Agile family of methodologies were applied in development, including the application of test-driven design and construction, literate programming and design-by- contract.

Development for QPS

One of my primary roles while with QPS was as Senior Analyst/Programmer and Technical Lead for large-scale Data Conversion and Cleansing Projects, mainly for their Polaris product. In this role I was responsible for the creation of a considerable number of fundamental tools and libraries, and project specific applications, characterised by their ability to manipulate extremely large data sets very quickly and efficiently. Beside coding and testing of these products, I was required to provide detailed program specifications and higher level design documents to junior staff to implement.

My role as Technical Lead in the domain of Database Administrator with QPS saw me overhauling many of the tools used by QPS for database administration, including largely rewriting their Ingres database replication product. Again, in this role, I was responsible for the detailed design of products to be constructed by junior staff.

Coding at QPS was carried out in a mixture of C, C++, Ingres ABF, SQL and Unix shell scripts, all under various versions of HP-UX Unix. A small amount of coding was performed for one project with Clipper on the PC network.

Development for DNR

As Database Administrator for IVAS at DNR, I undertook a large number of programming activities, in addition to assistance with the maintenance of the IVAS product itself . These coding activities included:

- overhauling the database management tool suite, resulting in a reduction in required database managers from two to less than one;
- creation of a sophisticated database integrity test suite;
- creation of a Linux-based solution, using a Web interface, to allow end-user support staff to create ad-hoc data extracts from IVAS and write them to magnetic tape for clients.

While with DNR I was seconded to the LGIP project as Analyst/Programmer. This required me to design and construct a suite of tools that could accept data provided by every Local Government Authority in Queensland (in a more-or-less arbitrary format), parse and cleanse that data, compare it to IVAS, then prepare update records to correct data within IVAS. This suite included various Unix based tools and a VisualBasic Windows tool to allow manual data manipulation. The Unix tools were distinguished by some sophisticated algorithms for parsing names and addresses out of more-or-less arbitrarily formatted input data sets.

Programming for DNR was carried out with C, SQL, Ingres ABF and Unix Shell scripts under Unisys DYNIX Unix. A mixture of C, Unix Shell scripts and JavaScript was used for Web-based products, and VisualBasic was used for part of the LGIP project.

Development for PRSC

My programming activity at Pine Rivers Shire Council was extremely broad, covering everything from dog registration systems through the rates and accounting systems through to the on-line library catalogue. Data access and manipulation systems were written in:

- MUMPS, initially DSM on PDP-11 systems, later MSM under HP-UX Unix;
- C, SQL, Ingres ABF and Ingres Vision under HP-UX;
- RSTS/E Basic on a PDP-11

Various small utilities were written for the PC network using a mixture of C and MS-DOS batch scripting.

Development for Shannon Robertson Systems

At Shannon Robertson Systems I was responsible for maintenance and development of two product streams. The first of these was a complex FeedLot Management product, running under MS-DOS, based around dBase III+ and written with the compiled Clipper language. This product managed all financial, feeding, and beast management tasks germane to the operation of cattle feedlots. The other product stream was suite of stockbook programs for cattle breeders, later extended to cover goats and sheep. This was developed in collaboration with the Australian Beef Research Institute (ABRI) at the University of New England to server as a remote front-end to their BREEDPLAN system. The program evolved into the HerdMagic product still marketed by Saltbush, the wholly owned subsidiary of the ABRI.

Database and Data Design and Construction

Beside considerable and broad experience as an Analyst/Programmer in designing, specifying and creating programs in a variety of procedural languages, I have a similar breadth of experience with database schema design and data design, including design of database schema and data definitions for the following:

- Person, Specimen Charges, Offender History, Narrative for Polaris at QPS;
- IVASe for DNR;
- Cash Receipting, a Workflow Automation system, and the associated varied license registers for PRSC;
- Stock Book systems for cattle breeders , and a feedlot management system, for Shannon Robertson Systems

My design activities for these schema involved a variety of tools, including Rational Rose and ERwin, and ranged from logical database design through to detailed physical design, creation of SQL scripts for implementation of the design, and design of the physical database to contain the schema. As part of the database design activities for IVASe, I was instrumental in the creation of a system, incorporating Visual SourceSafe, to provide a means of versioning the database schema. This system remains in use, and is recognised as a model solution to the problem by Ingres database administrators across Queensland.

Modern Approach

Over the last few years, I have actively explored and encouraged the adoption of new technologies and paradigms for the design and construction of IS solutions. I have particular respect for the strengths of the UML as a tool to assist in the design of complex systems. I also have respect for the various agile methodologies championed by the Agile Alliance, including Extreme Programming and Pragmatic Programming. While, to date, I have not been able to fully implement these paradigms, I have successfully implemented elements of them, and particularly attest to the power of a test-driven approach to programming. Fortunately TNSi is a fervent user of Agile methodologies, and while I have not been involved in the administration of the development process, I have had the experience of working within a fully mature Agile implementation.

I was able to fully apply a test-driven and pragmatic approach to the design and construction of the PES and Missing Persons DCC projects for Polaris at QPS, and strongly promoted their use to the current DNA project within QPS. Through these projects, I was able to demonstrate the significant cost benefit of these paradigms, particularly the significant reduction in defects and the associated costs in maintenance after implementation.

My experience has led to a firm belief that training and end-user documentation are a significant part of any developed solution, and that substantive expenditure by an organisation on these areas at an early stage significantly reduce long term costs. Further to that, I strongly push for good technical documentation and reasonably firm adherence to accepted code standards to aid in reducing maintenance costs, and am devoted to the credo that software should just work, out of the box.

Relational Database Technologies

The RDBMS with which I have the most familiarity with as a database administrator is Ingres, adopted by Pine Rivers Shire Council in approximately 1991 as part of a strategy to replace older data persistence solutions. I have worked with the entire suite of Ingres products with all versions from 6.3 through to II 2.0. In relation to Ingres I have performed the following activities:

- database administrator;
- database designer;
- database analyst;
- system and program designer;
- technical lead, including training and mentoring junior staff;
- programmer, using ABF, Vision, C, C++, SQL and various Unix shell tools.

All database activity within HPA involved the use of different versions (primarily 9i) of Oracle in production environments. While I was not responsible for implementing or maintaining Oracle instances, I was responsible for the design and construction of physical schemas within these instances, and am familiar and comfortable with the use of various Oracle client tools. For development and testing purposes I pioneered and promoted the use of MySql and Postgres.

Postgres is used throughout the TNSi Payment product, and I have worked against Postgres as a developer, with RDMS access brokered by Hibernate. In addition my engineering support role frequently required me to craft SQL to run against

production databases in order to diagnose faults, and to repair damaged data. The emphasis on these DDL and DML scripts was to minimise impact on production performance, reversability of modifications, and audit of modifications.

At PRSC, the legacy systems used ISAM and VSAM data files under RSTS/E, or more frequently MUMPS, and it was the latter which I worked with most. The data persistence model for MUMPS, although implemented as a BTREE with some additional links between data nodes, is not strictly speaking a relational database, although it supports the construction of a relational data model. The majority of the MUMPS-based databases at PRSC contained relational and semi-relational schema.

Prior to working at PRSC, my role as Analyst/Programmer at Shannon Robertson Systems saw me working with a number of PC-based database systems. This included various versions of dBase, particularly dBase III+, and relational and semi-relational database products built on top of the Novell BTRIEVE library.

I have had exposure to a range of relational database products, and am familiar with their characteristics and behaviour. These products include:

- Microsoft SQLServer 2000;
- Ingres 6.3 through II 2.0
- Sybase;
- Oracle 8, 9 and 9i, and 10G;
- Postgres;
- MySql versions 4.x and 5.x;
- FileMaker

Data Analysis

In fulfilment of all my roles through my career, including Analyst, Programmer and Database Administrator, I have been required to perform analysis of both business and systems operation data. Much of this analysis has been performed against significantly large or complex data sets.

I have applied the following tools in analysis of data:

- deriving data distributions and definitions of data domains;
- statistical analysis;
- decomposition and aggregation of tuple sets in relational databases using SQL;
- financial, reconciliation and auditing reports for both financial business data and computer system operations;
- construction of complex predictive models, such as beast growth models for feedlot management systems.

In addition to analysis of data sets, an important part of my efforts in system and database design has been the analysis of the nature of businesses data to be manipulated or stored. This has included discovery and documentation of data domains, business rules applying to data elements, and analysis of data flows through the enterprise. An example of this sort of analysis would be the exhaustive documentation of work flows related to issuing of licences and permits by Pine Rivers Shire Council, culminating in recommendations for significant Workflow Automation coupled with Document Imaging solutions.

Database Design

Having worked with database-centric systems for my entire career, and fulfilled the role of Database Administrator for approximately ten years, I am highly competent and comfortable in the realms of logical and physical database design, particularly in reference to relational databases. This includes design and specification of infrastructure such as:

- RDBMS server configuration;
- physical storage;
- client/server and distributed data service architecture;
- audit requirements;
- backup and recovery solutions.

I have been responsible for or heavily involved with logical and/or physical database designs for many significant products, including:

WORKPLACE	SYSTEM
HPA/Salmat	Tracking / Logistics System Authorisation System Scanning Service Qld Transport Forms Registry Online Assessment
QPS	Missing Persons DNA Person Offender History Specimen Charges Narrative Source of Information
DNR	IVASe
PRSC	Workflow Automation Dog Registry Cash Receipting Building Permits other licence registers
Shannon Robertson Systems	Feedlot management products Stock Books

Data Manipulation Programs

A considerable proportion of the programs that I have written over the past decade have been dedicated to the manipulation of large quantities of data, often transforming and transferring it from a data source to a data sink. These data manipulation programs are listed below with notes of their particular distinguishing features.

HPA Print Preparation

Java programs were developed to take large volumes of client data and transform it for printing through internal products. These programs were characterised by adhering to strict SLA time frames, and providing detailed auditing and reconciliation information. Due to the nature of the operating environment for these tools, they were designed to validate the incoming data stream, and then self-validate at key points through the processing pipeline, thus preventing erroneous data being committed to the printers.

Document Conversion Auditing

One task undertaken by HPA for a third party was the conversion of a very large number of physical documents to electronic form. There were complex requirements for audit and reconciliation of this process, related to billing, and to ensure the validity and consistency of the process. A combination of UNIX scripts, Java programs and SQL-based reports was constructed to deliver audit and reconciliation.

Remittance Processing Auditing

For a time HPA was delivering remittance processing for third party clients. This involved collection and processing of physical cheques and the formulation of XML deposit information submitted to the banks. Audit and reconciliation processes were developed and designed to ensure consistency between incoming and outgoing information, in parallel to and external to the remittance process itself.

Polaris Data Conversion and Cleansing

All Data Conversion and Cleansing (DCC) tools were distinguished by their adherence to very strict security, integrity, auditing and reconciliation requirements. They were written in such a fashion that, when combined with the logs and other outputs from their runs, they could be used in a court of law to unambiguously demonstrate the validity of their cleansing and conversion activities.

PROJECT	ACTIVITY	DISTINGUISHING FEATURES
1.8	Person Cleansing	Automatic reconstruction of broken relational structures and cleansing of invalid names.
	Specimen Charges	Sophisticated text parsing written with simple UNIX shell tools.
	Narratives	Sophisticated text reformatting written with simple UNIX shell tools.
Offender History	Cleansing & Conversion	Automatic reconstruction of broken relational structures; automatic cleansing of invalid places of birth; rapid merging of data sets from two legacy systems where the data sets were over 1 GB in size and contained millions of records.
PES	Cleansing & Conversion	Rapid transformation of very large data sets containing millions of records, including name parsing.
Missing Persons	Cleansing & Conversion	Rapid transformation of data sets from two legacy systems in semi-relational form to a fully relational form.

IVAS Data Conversion and Cleansing

While working for DNR with the IVAS and IVASe projects, I was responsible for creating programs for two data take-on projects related to IVAS. These were the Street Address Lot/Plan Index (SALI) and Local Government Interface Project (LGIP). Both of these projects required me to construct code able to parse street addresses from more-or-less arbitrary formats to a structured format suitable for import to IVAS. The LGIP programs were further distinguished by their ability to perform a similar function with property owner names and postal addresses, consistently producing over 90% accuracy. These tools were used to process data for all properties in Queensland, amounting to millions of records.

Pine Rivers Shire Council Financial Systems

I was instrumental at Pine Rivers Shire Council for maintenance of their rating and other financial systems. As well as maintaining the tools used for calculation of annual rates, I was required to produce a large number of ad-hoc statistical and financial analysis reports derived from the property and other financial systems.

Shannon Robertson Systems Statistical Reports

Both the feedlot management and stock breeding products I was responsible for at Shannon Robertson included significant reporting components. These spanned statistical analysis of beast growth patterns and cost of growth, simple income/expenditure modelling, and reasonably complex predictive models for beast growth and optimum breeding pairs.

Data Presentation in a Web environment

Most of the applications written while at HPA included a Web interface, usually for server administration and configuration. This interface was implemented mainly with JSF and Struts, but there was some static XHTML as well. In general data extracted from the servers was in XML form, which was transformed with XSLT into XHTML pages. Significant effort was made with these web pages, even though they were dynamically generated, to ensure they were compliant with World Wide Web Consortium standards and specifications for XHTML, HTML 4.0.1 and CSS 2.1. At the time of writing, a project is under way to move from Struts to the more modern JSF framework, including rich client components.

While at IVAS, I acquired the skills to provide a Web interface to:

- System monitoring and maintenance tools;
- An ad-hoc client tape production system;
- Interactive serving of Unix man pages.

This Web interface was implemented via an Apache server running on RedHat Linux, using a mixture HTML 4.01, CGI scripts written in C and as Unix shell scripts, and some Server Side Include (SSI) pages.

For over fifteen years I have maintained a personal Web site, currently located at <<http://partimepolymath.net/>>. I use this site as a testbed for mark-up and information design experiments, with an interest in maintaining strict compliance

with World Wide Web Consortium standards. The current version of the site conforms to strict XHTML 1.0 and CSS 2, and includes use of JavaScript for a variety of purposes. I have also constructed and maintained web sites for the various organisations I have been involved with, usually including forums, Wikis and various PHP/MySQL based applications.

I have a particular interest in acquiring skills in one or more of the modern “scripting” languages, particularly LUA, Ruby and Python, as a means of simplifying the transformation of complex data for presentation of data in a Web environment.

System Management and Database Administration

I have held the role of Database Administrator for these organisations:

1. Queensland Police Service (QPS), Information Systems Branch, December 1998 – August 2003;
2. Queensland Department of Natural Resources (DNR), IVAS and IVASe projects, October 1995 – December 1998;
3. Pine Rivers Shire Council (PRSC), circa 1991 – October 1995.

For each of these organisations, my fulfilment of the role of Database Administrator went beyond day-to- day Relational Database Management System (RDBMS) maintenance and encompassed

- Capacity planning and resource costing as it related to current and projected use of databases, including human resource costing;
- Performance monitoring and analysis, often from the point-of-view of determining cost-benefit of current systems and proposed alterations;
- Information Security and System Access control, encompassing audit and reconciliation concerns and system and data recoverability and validity;
- Database schema and RDBMS product change control;
- Integration with application and infrastructure change processes;
- RDBMS product maintenance and upgrades.

My roles as Senior Analyst Programmer and Technical Lead with QPS and PRSC required me to undertake system management activities, either solely or as part of a multidisciplinary technical team, depending on the project in question. These activities primarily required resource monitoring and system analysis in the following areas:

- Unix and network performance;
- Estimates of resource costing, including human resources;
- Change management as related to application change, but encompassing the application infrastructure;
- Infrastructure capacity planning.

Architectures

The majority of the information systems that I have designed, constructed and maintained have been database-centric.

The core TNSi Pay systems are comprised of a large number of loosely coupled services mainly communicating via JMS with rigorous separation (and firewalls) between the demilitarized web-facing tier, the display layer tier, the services tier, and the physical database. Database access is via JDBC. Some services communicated via HTTP, and some by direct socket reads and writes - this was also the case for most outgoing communication. Customer facing interfaces were mainly delivered via HTTP, either as REST web services or web applications delivered through a browser. The system was further complicated by being deployed across three sites around the world, with multiple hot- and warm-swap hosts and service instance within each site.

The legacy systems I have been involved with, and the systems I constructed in the first years of my career, had a simple 2-tier architecture, with the application communicating directly with a data persistence layer present on the same host computer. Although IVAS and the Ingres systems developed at PRSC communicated to the Ingres database servers via the Ingres/Net communication product, this was mainly a method for partitioning large logical databases into smaller physical databases.

The bulk of the systems developed at HPA were full-blown J2EE compliant Web Services. These were provided with XHTML based management and configuration interfaces generated through Struts and JSF, and the Web Services themselves were expressed through Axis. Companion client libraries for Java development were created, but the major-

ity of clients were written in C# under .Net. All these services were explicitly designed to be loosely coupled to each other, although they shared considerable common fundamental code.

IVASe at DNR and the Polaris system at QPS were more complex 3-tier client-server systems. The former used a Visual-Basic presentation layer on Windows PCs, with Tuxedo as a transport layer communicating with Ingres as a data persistence layer. Polaris used a suite of simple Ingres ABF clients which communicated via Ingres/Net to the data persistence layer, but separated the application servers from the data servers to improve system reliability and partially alleviate single point of failure concerns. This was assisted by maintaining a shadow copy of the primary database that could, theoretically, be used in a warm-swap recovery in the event of failure. Polaris also made some use of Tuxedo and MQSeries to provide gateways to legacy systems and external data sources.

While at DNR and QPS, I made use of distributed operations and multithreading in the design and construction of small products and ancillary support programs, even though the primary applications at the site did not make use of these technologies. Examples of this are the Ingres Data Replicator and Database Comparison tools used at QPS, which distribute their operations across several hosts, using a mixture of RPC and remote shell operations. Another example is the conversion tool I constructed for the Missing Persons project, which was optimised to reduce total run-time, at the cost of increased CPU load, by significantly multithreading various phases of operation.

The web-based tools developed for IVAS (an ad-hoc tape generation system, and simplified interfaces to some IVAS systems maintenance activities) utilised the technologies common in the late 1990's, i.e. dynamic pages constructed with CGI and SSI, and HTML form elements for data input. I have a personal interest in exploring and exploiting the significant power inherent in newer Web technologies, particularly SOAP, RDF and RSS.

I have been involved with GIS systems at two locations. While at IVAS, I was exposed to the BLINMAP project through initial testing against and integration with the IVAS system. At Pine Rivers Shire Council, I was heavily involved with the introduction of GIS to the site, particularly with the development of various bridges between the GIS system and the Property data held in legacy systems.

Information Systems Planning

I have long held the opinion that planning, as it relates to Information Systems, is an integral and vital part of the implementation and maintenance of any system. Inadequate planning prior to embarking on any system change has been demonstrated quite comprehensively to be one of the greatest risks to successful and cost-effective system change implementation. My awareness of the importance of planning at all levels of IT activity has led me to strong experience with several different planning domains:

Project Planning

One of my primary roles while with QPS was as Senior Analyst/Programmer and Technical Lead for large-scale Data Conversion and Cleansing Projects, mainly for their Polaris product. In this role I was responsible for the development and management of project plans for several major projects, namely:

- the almost-complete replacement of the existing product in 1999 with a new product referred to as Polaris 1.8;
- the cleansing and data take-on into Polaris of Offender History and Microfilms data from legacy systems;

I was also responsible for the creation of the data cleansing and taken-on project for Missing Persons data from several legacy systems.

The project plans for these data cleansing and take-on projects not only covered the specific activities for cleansing and take-on, but also included the design, development and implementation phases for relevant tools and environments. These projects involved between two and five key resources, and required the co-ordination and involvement of various service providers external to the project team.

Through 2001 I was technical lead for a project to upgrade all Ingres installations across QPS to the same product level, in preparation for a corporate wide operating system upgrade. In this role I was responsible for providing advice to the project manager, and had significant input to the project plan, particularly for the design of analysis and testing phases and identification of associated required changes. I was solely responsible for the creation of detailed plans for the upgrade each of over 20 Ingres installations, and the planning required to significantly overhaul the management processes for Ingres installations at QPS.

I have made frequent use of various Project Planning software products over the past 23 years, including most versions of Microsoft Project. I am very comfortable with these tools, and have often used them to produce Gantt and Pert charts,

and for resource costing and time planning, for large scale projects as well as smaller ad-hoc activities, including programming for the annual Abbey Medieval Festival.

Change Management

Having fulfilled the roles of Analyst/Programmer and Database Administrator, I have frequently been involved in the creation of solutions which required changes to existing systems. I have also frequently been called on to implement required changes developed by other team-members or external agencies.

Generally the implementation of system changes has required careful planning, and one of my responsibilities at HPA, QPS, DNR and PRSC has been the creation of suitable plans. The extent and complexity of these plans has varied from site to site, and from project to project. Frequently the planning has simply required:

- impact analysis;
- scope definition;
- analysis and definition of relationship to other changes;
- resource and time allocation;
- integration with change management and release cycles.

On occasion more complex planning has been required. An example of this would be the Ingres II upgrade project performed through 2001, which entailed the co-operation and assistance of a significant range of resources external to the project. A complete, detailed project plan was developed, utilising Microsoft Project for project design and tracking, supported by a considerable number of detailed implementation plans for specific steps.

Another example would be the upgrade of the Polaris National Names Index extract system for QPS. This is a system where data is extracted from Polaris, merged with data from a legacy system, then forwarded to the National Names Index (NNI, previously known as CrimTrac). As a result of a significant overhaul of the product, the volume of data obtained from Polaris increased roughly tenfold. It was necessary to plan for alterations to database schema and contents, introduction of new versions of software, decommissioning of old versions, exhaustive testing in collaboration with the external agencies, initial take-on of a significantly large data set by NNI, and elaborate regression paths. Planning was further complicated by the need to integrate these changes with the tri-weekly Polaris release cycle and cross-dependencies with other Polaris product changes. This mini-project was successfully completed in June 2003, and has significantly improved the extent and quality of data available for nation-wide criminal investigation.

Fortunately TNSi had a reasonably mature change management system, which allowed the relative luxury of releasing software changes either through planned releases timed to fit into very limited downtime windows, or as emergency patches. In either case while responsibility for the implementation of changes ended at the point where artefacts and release or installation instructions were delivered to operations staff, I invariably took on the additional responsibility of monitoring the installation process and verifying post-release the success of the change.

Infrastructure Planning

A fundamental component of change management as it relates to RDBMS and database schema changes is analysis of, and provision for, changes in data volume and data access patterns. I have frequently been required to plan for, or contribute to planning for, system infrastructure changes. This has included specification of and planning for changes to:

- data storage hardware;
- audit and reconciliation systems;
- server security and data access control;
- data backup and recovery;
- server hardware and software;
- RDBMS products;
- networks;
- data access client software.

Beside these general RDBMS-related infrastructure planning activities, I have been responsible for specific infrastructure planning and specification for a number of products:

- Data storage, manipulation and interchange infrastructure for the LGIP project for DNR, encompassing the provision of PC and Unix software;

- Specification of new Unix servers for the IVASe project for DNR, resulting in the purchase of several high-end Sun systems to replace two Unisys hosts used by IVAS;
- Preparation of budget, organisation of quotes and tenders, and purchasing for IT expenditure for PRSC in 1995, including a complete replacement for the existing PC network.

One of the most extensive and complex infrastructures I have planned and implemented was the Data Cleansing and Conversion (DCC) arena for QPS. This required not only planning for the provision of physical data storage but also for the creation of fundamental tools and libraries, security and reconciliation protocols, and comprehensive programming and data handling guidelines and standards.

Support

I have been involved in a broad range of system support scenarios. This has included direct end-user support, support at client sites, remote support, technical support in-house and second-line support.

Having a National R&D role within HPA meant that the bulk of support I provided was second-line in-house support to state branch IT groups, and to other members of R&D, but there was some support direct to operations groups around the country for certain applications.

The explicit role I held within TNSi was designated as Continuous Engineering, and our responsibility included second-line support for a complex production environment. I seldom dealt directly with our ultimate end-users (who were a mixture of merchants, users of merchants' web sites, banks and other payment service providers), but instead dealt on a daily basis with our Customer Support staff. Frequently dealing with the support issues surrounding a complex system constructed of many loosely connected components required the ability to quickly distinguish between emergent behaviour experienced by the end user from internal root causes.

Queensland Police Service systems are, on the whole, 24x7x365 systems and consequently have fairly elaborate support arrangements. My primary support responsibilities here were second-line technical support for the Ingres product. While not required to be on-call at all times, there was an expectation that I would respond to crisis situations wherever possible and be willing to come into the office when required. I fulfilled this expectation whenever the need arose, demonstrating my willingness to put my shoulder to the wheel in times of crisis. I was seldom called on to liaise directly with end-users while at QPS, but I was required to liaise closely with first-line support staff and business representatives in the resolution of faults.

In addition to business-systems support, I was responsible, in my role as Technical Lead, for supporting other staff both in their use of the Ingres product, and in use of the Data Cleansing and Conversion arena I had been responsible for designing and constructing.

The IVAS system at DNR was not a 24x7 system, but instead operated in an on-line mode during business hours with the other hours devoted to intensive batch processing and system maintenance. As a result of this I was required to be on-call 24x7 to resolve system failures when required. As with QPS, I was not responsible for direct support of end-users, but instead provided support via a team of business representatives. In addition I provided support to Analyst/Programmers in using the Ingres product, and guidance in developing software to operate against the Ingres RDBMS.

At PRSC I was part of a team of IT specialists whose role included direct end-user support. While the bulk of our users were at the main branch office, some were at remote sites, and a substantial part of the support was performed remotely or via visits to the remote sites. End-user support encompassed support for the use of in-house systems, off-the-shelf packages, and hardware.

This was similar to the regime at Shannon Robertson Systems, who provided software and hardware for the agricultural market. The bulk of our support was provided remotely, including visits to remote sites and utilisation of remote-desktop products. Considerable support of the sales team was also required.

Problem Solving

When mentoring junior staff, I make strong efforts to instil in them the same problem-solving and analytical approach that guides my own activities as an experienced IT professional. I emphasise the importance of clear analysis to gain an accurate and complete understanding of a problem, before attempting to resolve the problem.

Having a long history in the IT industry, I have a strong interest in improving the processes and tools used to perform my work. This has led me to keep abreast of emerging technologies and philosophies, and actively promote their adoption wherever appropriate.

This has ranged from my exploration of and experimentation with elements of Extreme Programming (XP) and other Agile Methodologies, through generating recommendations resulting from analysis of the benefits of adopting Open Source technologies, to formal evaluations of specific products.

This history of critical analysis of and reporting on emerging technologies stands beside the significant quantity of requirement and performance analysis I have performed for specific systems and programs as part of my roles as Analyst/ Programmer and Database Administrator.

As my tertiary training included a study of Physics, I generally apply the scientific method to the resolution of problems in the arena of information technology:

- collect observations and measurements;
- form a predictive hypothesis to explain or model those observations and measurements;
- test the hypothesis;
- iterate until a solution is obtained.

Having been trained in this sort of rigorous problem solving methodology has instilled a preference and respect for clear and precise problem and solution definition, informed by formal principles. I have found two of these particularly germane to professional practice in Information Technology, namely Hippocrates' dictum *Primum Non Nocere* ("first, do no harm") and Occam's Razor, advocating choice of the simplest complete explanation and solution.

My grounding in these formal methods may be one reason for the attractiveness of the Agile Methodologies such as Pragmatic Programming and XP, which strongly advocate methods which lead to precise, minimal solutions in a cost-effective and timely manner.

Skills Acquisition

Over my more than 24 years as an IT professional, I have taken it upon myself to acquire a wide variety of IT skills. As an example, Java, Struts, JSP, C, C++, XHTML, XML, CSS, JavaScript and AppleScript were all self taught.

My Ingres abilities, and Ingres ABF skills, were successfully acquired through formal training courses, and significantly expanded on by self-guided research and reading. Other language skills were successfully obtained either through tertiary training or instruction by peers and mentors.

My success as an IT professional clearly demonstrates my significant ability to quickly and successfully acquire new skills

Technology Evaluation

I have performed two different types of Information System evaluation. The first has been the evaluation of the suitability for externally sourced products and solutions to the problem under consideration. The second has been the evaluation of existing systems, from a variety of view points.

My role in R&D with HPA/Salmat explicitly includes a brief to remain aware of and evaluate new and developing technologies in my scope of activity. This has led me to increase the use and adoption of open source solutions within the organisation, and to promote the adoption of J2EE 1.5 methodologies over 1.4.

The bulk of product and external solution evaluation I performed while at QPS was secondary activity related to the main thread of a project. I was responsible for submitting a variety of reports to senior management advising on the suitability and effectiveness of:

- Rational Rose for small systems design, rapid systems deployment and database design;
- C++, considering the skill base and experience within the Information Systems Branch of QPS;
- XP and other Agile methodologies for rapid systems deployment;
- Validity of Ingres II 2.0 under HP-UX 10.20 and 11.0;
- adoption of open-source products such as Doxygen and Jam

I was similarly responsible for a broad range of product evaluation at PRSC, particularly while assisting with the development of annual purchasing, generally as part of the process of tender evaluation. This included detailed evaluations of personal computers, laptops, dumb terminals, printers, network operating systems, RDBMS products and database management products.

Requirements and Applicability Analysis

The other realm of IS evaluation has been examination of existing systems. While a significant part of this has been part of the analysis activity in the design of replacement systems, where the scope of examination has emphasised the functional aspects of the system, I have also been required to evaluate the run-time behaviour of complex systems. Sometimes this has been to trace system faults, and other times this has been in order to identify ways in which a system can be made more efficient. Generally these evaluations have relied on a number of metrics, with special attention paid to:

- transaction rate;
- CPU load;
- I/O load;
- memory and disk space requirements.

Examination of systems in terms of metrics such as lines of code, screen counts or function point analysis has been of low importance.

System Development Lifecycle Knowledge

Having worked as an Information Technology professional since 1988, I have kept abreast of industry best practices as they relate to the design and construction of large information systems organised around databases, both relational and non-relational.

Much of my experience has been informed by working environments dominated by fairly traditional development approaches, and where there was a mixture of new development and maintenance of legacy systems. I have become a keen proponent of agile methodologies, and of the desirability of obtaining solutions, rather than constructing them, wherever appropriate. The open source community provides a wealth of tools for reuse, and the Pattern Language community similarly provides a wealth of inspiration and guidance for construction of new tools.

Project Lifecycles

The majority of the projects I have been involved with have followed a classical System Development Life Cycle (SDLC) model, or some variation of that model according to local preference and practices. This model, consisting roughly of the phases:

- requirements gathering and analysis;
- system design and specification;
- program and test specification;
- unit testing, integration testing, and user acceptance testing;
- implementation;
- maintenance.

has been successful in delivering large projects I have been involved with. These include IVASe at DNR, the Polaris 1.8 and Offender Management systems at QPS, the Ingres II 2.0 upgrade at QPS, and the implementation of the Workflow Automation systems and related components at PRSC. On the other hand, industry concerns that this classical model is not responsive to changing requirements and can be slow to deliver a product are justifiable. The very long lead time for the development and release of the original IVAS product is a good example of this.

While at QPS, I was exposed to initial experimentation with the Rational Unified Process (RUP), primarily through the provision of database designs during early iterations. Over the last few years I was at QPS, I took the opportunity to experiment with Agile methodologies in order to implement components within the larger SDLC framework controlling the overall project. The rapid design/build cycles characteristic of these processes, with a de-emphasis of the long initial specification and design phase, proved to be extremely cost effective for the PES project, where a complete suite of data cleansing and conversion tools was able to be delivered within two weeks of the initial statement of requirement.

A particularly interesting side-effect of the experiments with rapid design/build cycles while constructing the suite of foundation classes for a re-implementation of the Polaris DCC arena was that there was very close correlation between the requirements and design and the constructed code. This significantly simplified code maintenance and enhancement, and the presence of accurate and up-to-date technical documentation at the completion of each cycle encouraged the maintenance of that documentation to reflect changes in subsequent cycles.

Working within a more loosely structured R&D group at HPA, and being only one of two developers working on client-side J2EE solutions, meant that elements of the Agile methodologies were even more applicable. While it was not ideal

that we were responsible for both design and implementation, certain elements of these methodologies still promoted quality, particularly the use of test-driven development, literate programming and design-by-contract.

The design and implementation activities within TNSi are firmly wedded to Agile methodologies, but with the complication that continuous deployment is not currently possible - as the systems are sold as 99.999% uptime, there are limited windows for production updates, and the final release stages from user acceptance testing onward resemble more traditional processes. Work is progressing to move toward a continuous deployment environment.

Implementation

Aside from the creation of a suitable project plans, and monitoring progress of that plan, there are two phases in the implementation of an IS solution. I have noted above my extensive experience in the first phase of developing a solution through the design process to instantiating a solution as concrete products that can be shipped. The phase subsequent to the apparent provision of a final product is the introduction of the product to its run-time environment.

Through a mixture of code-review processes, design-by-contract and test-driven programming, I have always endeavoured to produce defect-free solutions. Of course the reality is that there are inevitably bugs in apparently finished programs. To address this, I have generally relied on a staged release environment. At QPS and DNR, completely elaborated staged environments were available, allowing the use of a rigorous change management process that saw the progression of all changes through Integration and Acceptance Test environments after they left Development and before they entered Production. PRSC relied on an Acceptance Test environment alone, with integration testing occurring as part of the development phase of activity.

Within HPA I pioneered and promoted the adoption of a multi-stage release environment, and helped move the R&D group to releasing complete defined versions of applications as packages which included documentation and installation mechanisms.

Configuration Management

I perceive the discipline of Configuration Management to have three primary facets:

- maintenance of catalogues identifying all configuration items, including documentation of the physical and functional characteristics of those items;
- recording and auditing changes to the characteristics of those configuration items;
- recording and auditing change processing resulting in changes to the characteristics of configuration items.

Wherever possible, I have integrated Configuration Management processes and policies tightly with Change Management processes and policies, as the two disciplines strongly inform one another.

Beside maintenance of configuration and assets registers, I have made use of various tools to simplify configuration management, including:

- use of staged release cycles, progressing configuration changes through testing and deployment environments;
- version control of database schema, control data sets and program code;
- establishment and enforcement of Standard Operating Environments (SOE).

Communication Skills

The principles of effective communication were integral to my formal training as a teacher, and have been strengthened by subsequent experience. I have a personal interest in the formal principles of design and communication, and the application of good design principles to effective communication.

I communicate such that:

- messages are designed and delivered to gain the attention of the intended recipient;
- I clarify and check that the message intended was the one received;
- I take time and fix my attention in listening to others, making time for them to complete their messages;
- I search for meaning and messages both said and unsaid in order to ensure that I fully understand the intended message;
- I try to give clear and unambiguous feedback;

- I am sensitive to an understanding of the emotional and vested-interest content of messages, as these are frequently the true motivators for people's actions.

My written communication ability can be judged by resources such as the current document and the varied writing at my personal Web site. It can also be judged by the successful provision of a considerable amount of technical and end-user documentation.

My oral communications ability is demonstrated by my experiences in the provision of formal and informal training, and in by the successful completion of significant system analysis and requirements gathering activities.

At each workplace I have held the respect of my peers, and have received expressions of high regard by senior management. I have frequently been trusted with the management of sensitive information.

Negotiation

Negotiation is the process whereby each party has an initial position, evaluates the other party's and adjusts their position in light of this evaluation. In developing negotiation skills, my experiences and reading has lead me to the understanding that the key phases and tasks of negotiation are:

Preparation: prior to the negotiation I obtain information on:

- the wants and needs of both parties,
- the likely negotiation style and tactics of the other party,
- the other party's likely assumptions,
- the best choice of timing, location and agenda,
- the best way to formally assess and record progress, and
- timing constraints impacting on the decision.

Familiarisation: to build rapport or relationship;

Proposal phases: the parties express wants including 'must gets', 'should gets' and 'could gets', and both stated and unstated positions are 'interrogated' with persistent and probing questions until clear positions are established;

Bargaining in which:

- conditions are stated and I am specific about what I expect in return, and
- concessions are traded, with deadlocks or impasses overcome by either giving away concessions (instead of trading them), or moving onto a higher level of arbitration.

Agreement phase in which parties with authority may agree to broad or specific positions.

The best demonstrations of my negotiation abilities lie in the successful completion of a great number of significant projects which required co-ordination of the needs and requirements of a large and diverse pool of stakeholders and service providers.

My abilities in this arena are also demonstrated by my participation in budget preparation and fulfilment through tender processes while at Pine Rivers Shire Council. This required careful balancing of budgetary and technical constraints and end-user requirements within fairly short timeframes.

Written Communication

Industry best practice, and my own experience, clearly identifies that inadequate documentation frequently creates unnecessary costs in system deployment and maintenance. To this end I consider appropriate and comprehensive documentation and training to be a component part of any IS solution.

Documentation should include more than a simple technical description of the characteristics and behaviour of delivered products. It should encompass the rules, guidelines and procedures germane to those products. Documentation describes the culture in which the products will operate, and forms that part of the delivered IS solution which connects users, operators and support staff to the products to form a complete system.

Technical Documentation

In the preparation of technical documentation, I have a strong preference to provide clear links between the various design and specification documentation, right through to the instantiated code. This means that I endeavour to ensure that, for example, product specification documents refer back to the design documents that caused their creation, and refer forward to the code modules constructed. By providing these cross-references, the effort of tracing back the reason for a particular implementation decision is much reduced.

When creating a product, the minimal documentation I develop to accompany it is a description of usage, with accompanying examples where necessary, a description of the role, source and function of the product, and notes on maintaining the product. Depending on the audience and purpose of the developed solution, I will also create a range of additional documents, including end-user documentation where required.

Over my 24+ years in the IT industry, I have created a considerable volume of design and specification documentation. Recent examples of this for QPS would be the DCC design documents, where I was responsible for preparing:

- process design and specification documents;
- program specifications;
- unit, integration and system test specifications.

These specifications were prepared for the 1.8, Offender History, Microfilm, PES and Missing Person projects. Similar suites of design and specification documents were prepared for the LGIP and Street Address-Lot/Plan Index (SALI) projects at DNR, and for the Workflow Automation and associated license register systems at PRSC.

At HPA I wrote technical documentation to accompany the Authorisation, Remittance Processing and Logistics systems that I was responsible for or involved with creating. This documentation included instructions for installation, detailed descriptions of the web and Java library APIs, usage instructions, and general overviews of the products. The Online Assessment project made heavy use of the Confluence CMS at my insistence for all design, implementation and standards documentation, and was intimately tied to the Jira defect management system and Subversion source control repository to ensure a very high level of transparency to the development process.

In recent years I have developed a strong preference for relating in-code documentation back to the design documents which inform it. More importantly, inspired by Donald Knuth's *Literate Programming* recommendations, I endeavour to provide considerable in-code technical documentation. The evolution of tools akin to JavaDoc encourage this approach, as they significantly simplify creation and maintenance of this documentation.

I have a preference for creating documentation in HTML, and a fondness for Doxygen for managing JavaDoc style technical documentation in C and C++, but will write using whatever tools are available and mandated by site standards.

Examples of HTML documentation I have produced, partially with the aid of Doxygen, include:

- Logical, Physical and Foundation class documentation for the QPS DCC arena;
- Project specific code module documentation for the QPS DCC PES, MissingPerson and DNA projects;
- Design and Specification documents for the QPS Microfilm DCC project.

In addition to these, most of the smaller informal guidelines I have written have been in HTML format.

An example of a more complex HTML-based documentation suite would be the documentation of the policies, procedures, guidelines and toolset for Database Administration for IVAS. This was an exhaustive documentation resource spanning dozens of pages, and included interactive HTML serving of Unix man pages.

My strong preference for HTML based technical documentation is informed by two prime considerations. Firstly, as a non-proprietary format, HTML is widely accessible by display devices ranging from all makes of personal computer, through dumb terminals to mobile phones and PDAs. Secondly HTML is an ideal format for documents which may be placed in a repository and be required to be accessible and readable for an indefinite and arbitrary time into the future.

Technical Standards and Guidelines

In my roles as Technical Lead, and in fulfilment of my responsibilities for mentoring junior staff and providing peer support, I have created and contributed to a considerable number of formal technical standards and guidelines. Some of those that I have created include:

- HPA JBoss and Log4J standards;

- HPA AuthSystem Best Practices;
- QPS Ingres Operations Manual;
- Polaris Code Review Process;
- Polaris Code Design Guidelines;
- Polaris Code Specification Standard;
- DCC Programming Standard;
- DCC Documentation Standard;
- Ingres IMA Guideline and Overview;
- Shell scripting guidelines and standards for IVAS;
- Ingres C and ABF coding standards for PRSC.

I have also written a considerable number of smaller informal documents, including such things as an introduction to Korn shell I/O redirection, Ten Commandments for writing C code, and Ten Commandments for writing Shell Scripts. Small informal documents like this have usually been generated to support peer training and mentoring activities.

End-user Documentation and Manuals

I have produced end-user manuals over my career, including manuals for these systems:

- IWS Server products, and the Tracking Developers Manual for HPA;
- IVAS ad-hoc tape system;
- PRSC Workflow Automation system;
- Various licence registries for PRSC;
- Product installation and operation manuals for the stock breeding and feedlot management products from Shannon Robertson Systems.

The volume of end-user documentation I have produced has been less than the volume of technical documentation, primarily because I have frequently been fortunate enough to work in teams which included specialists responsible for creating end-user documentation and manuals.

I have been responsible for the creation and presentation of a large number of briefing documents for management and clients related to the various projects I have been involved in specifying, managing or constructing. Generally these documents have constituted project and process overviews, but have included incident reports, results of fault analysis, summaries of tender responses, business option papers, and sundry documents to express technical issues in terms of the business domain.

End-user training

Prior to entering the IT industry, I trained and worked as a Secondary School teacher, teaching Mathematics and Science. I have retained my status as a registered teacher, and make efforts to remain current with educational theory. This includes involvement with the educational program of the Abbey Museum of Art and Archaeology.

My professional teaching skills have been applied to formal end-user training for HPA/Salmat, Pine Rivers Shire Council and Shannon Robertson Systems, training clients and staff in a variety of products and applications.

Mentoring

As Technical Lead in the realms of Java, C++, C, Ingres ABF and shell script programming, and in the realm of Ingres database administrator, I have frequently been called on to adopt the role of mentor to junior staff. My experience as a teacher make this an enjoyable activity.

My approach to mentoring involves a mixture of ad-hoc training, provision of documentation and suggested reading, and delegation of tasks suitable to the student's skill levels. I strongly encourage active learning and the adoption of a self-guided problem solving approach.

Professional Representation

In my role of Database Administrator, I have represented the interests of QPS, DNR and PRSC through meetings of the Queensland Ingres Users Group. In that role, and in the role of Analyst/Programmer, I have also been required to represent the department, section or team that I was working in to other departments or sections in formal forums.

Additional Experiences

My experience and training as a teacher required strong communication and negotiation skills, including involvement with professional forums.

I have a strong interest in medieval history This has led to my involvement in both the Richard IIIrd Society, for a time as President the Queensland branch, and with Living History societies, including a re-enactment group that I established in 2001. I have been Treasurer of the Queensland Living History Federation on several occasions and for many years was a committee member for the Apple-Q Inc Apple users group. Since 2005 I have been very heavily involved in the organisation and management of the annual Abbey Medieval Festival.

Through High School I was involved with a variety of leadership and public speaking programs, including becoming a regional finalist for the Lions Youth of The Year, and participation in ToastMasters International functions. These experiences instilled a confident and relaxed manner when presenting complex materials in formal and semi-formal settings. Through 2003 alone I have presented historical papers to the Australasian Ricardian Conference, the Richard IIIrd Society, and at the annual Abbey Medieval Festival.

As a result of my involvement in these sorts of organisations, I am very experienced and comfortable in:

- communication with and education of a diverse audience;
- presentation of complex materials verbally and in written form;
- theatrical presentation;
- representation of the interests of my groups to external organisations.

Team Member

Most of my time in the IT industry has been spent as part of small to medium sized multidisciplinary teams. On occasion these teams were permanent organisations, while at other times they were ad-hoc groups formed for the completion of particular projects. These groups have encompassed a broad range of disciplines, personalities and cultures. Having worked closely with a wide variety of IT professionals and professionals from other disciplines, I take a strong cross-disciplinary approach.

I have engaged in information systems development, deployment and support as a member of these teams:

EMPLOYER	TEAM COMPOSITION
TNSi	Analyst/Programmers, Systems Administrators, System and Network support engineers, Customer Service staff.
HPA / Salmat	Analyst/Programmers, Web Designers, Systems Administrators, Project Managers, System and Network support engineers, Customer Service Officers, Sales staff.
QPS	Database Administrators, Analyst/Programmers, System Designers, End-user representatives (serving Police Officers), Trainers, UNIX System and Network support engineers, Project Managers.
DNR (IVAS)	Database Administrators, Analyst/Programmers, Administration staff, End-user representatives (Land Valuers)
DNR (IVASe)	Analyst/Programmers, System Designers, Technical Writers, Project Managers
PRSC	Analyst/Programmers, Hardware and Network support engineers, Technical Writers. Particular projects included appropriate end-user representatives ranging from dog catchers to building inspectors to librarians.

EMPLOYER	TEAM COMPOSITION
Shannon Robertson Systems	Analyst/Programmers, Sales staff, Dispatchers, Help Desk operators, Hardware engineers.

For the Missing Persons and DNA projects at QPS, I was involved in some experimentation with Pair Programming. My roles as technical lead and mentor has frequently required me to work closely on a one-to-one basis with other staff in order to successfully provide IS solutions.

As well as being able to function as a member of a team, I have strong abilities to work alone and self-direct effort. When working in this fashion I use the same skills of effective time utilisation, prioritisation of tasks, and stake holder communication.

Team Leader

In addition to leading small ad-hoc and short-lived teams formed for the resolution of particular problems, I have had the experience of managing the Information Services section at Pine Rivers Shire Council for six months. During that time I was responsible for

- technical supervision while encouraging self direction;
- establishing division of work;
- delegation of work;
- organisation of resources;
- facilitating financial management;
- giving feedback to keep on track and to motivate.

My roles as technical lead and mentor saw me apply similar disciplines in supervision of junior and less experienced technical staff in their professional activities.

Through the duration of the Online Assessment project within Salmat, I have performed the role of technical lead, providing an interface between the project management, testing and development sections of the project group. In addition I have had the responsibility for oversight of developers to promote and ensure quality and adherence to standards and specifications, with an emphasis on agile and traceable development activity through the use of CI and tools like Subversion and Jira.

I have been responsible, particularly at PRSC, for significant input into management functions generally associated with team leadership and management of small to medium IT groups, including:

- budget preparation;
- preparation of tenders and requests-for-offer;
- sourcing and selecting contract staff;
- recruitment and employment of permanent staff;
- hardware and software purchasing;
- review and acceptance of tenders and requests-for-offer;
- management of services provided by external agencies;
- formal performance appraisals.

My professional training and experience as a Secondary School teacher, and my experiences as committee member and President of various voluntary organisations have fostered skills which are transferrable to the domain of team leadership of IS groups:

- conflict resolution;
- crisis management;
- active and interpretative listening;
- empathy and encouragement in order to bring out the best in people;

- objective decision making and action planning;
- team building;
- resource management;
- task delegation and allocation.

Through example and encouragement, I try to foster a culture of respect for other team members, and am a strong advocate of handling trusted information with care. In the face of negative comments about individuals, I have frequently advocated for recognition of their good work and positive attributes. In face of conflict or misunderstandings within project teams between professional disciplines, I have encouraged dialogue between disparate views (for example resolution of time and cost constraints between hardware and software specialists). I maintain good general management techniques to keep team members informed of time, cost and quality considerations in relation to the work at hand. In all project work, I regularly seek updates for project progress and advise team members of this. I seek customer feedback, and relay this to my team. I support staff in the matters that affect their employment, and aim for a flexible approach. I offer encouragement and, where possible, reward for good work, and take efforts to ensure notable effort is widely recognised. I take the time to congratulate and thank team members, and project contributors, for good work, and support deserved career advancement however possible.

For each place of employment, I have made efforts to familiarise myself with the relevant human resources principles and practices, and to work within the confines of legislation and directives, equal employment opportunity, anti-discrimination, workplace health and safety constraints, performance management, and recruitment and selection procedures.

Project Management

I have undertaken project management activities for a number of projects implemented by small to medium sized teams, including

- the almost-complete replacement of the existing product in 1999 with a new product referred to as Polaris 1.8;
- the cleansing and data take-on into Polaris of Offender History and Microfilms data from legacy systems;
- the corporate wide upgrade of Ingres for QPS;
- the installation of a corporate-wide Novell network at PRSC, including the provision of new server hardware and software, and end-user documentation and training;
- the implementation of all infrastructure improvements provided for by the 1995 Information Services budget for PRSC, including outsourcing of some activities to external service providers.

I have made frequent use of various Project Planning software products over the past 23 years, most frequently various versions of Microsoft Project. I am very comfortable with these tools, and have often used them to produce Gantt and Pert charts, and for resource costing and time planning, for large scale projects as well as smaller ad-hoc activities.

In management of projects for small teams, I make use of other skills and tools as appropriate, including:

- frequent communication with team members and stake holders to track progress and issues;
- comprehensive documentation;
- continually updated issue and outstanding task lists;
- publication of project plans, timelines, key dates and progress;
- project diaries;
- formal and informal meetings and briefings in a wide variety of formats.

Knowledge of Quality Management

Having worked within State and Local Government for approximately 14 years, I am well grounded in the areas of service management, quality management and testing, and their relationships to each other.

The organisations I have worked with have spanned a range of maturity levels within the Capability Maturity Model (CMM), and different parts and aspects of the organisations have attained different levels. A significant part of my activities as technical lead has been the promotion and implementation of organisational change to encourage attainment of higher levels of adherence to the Capability Maturity Model. In terms of the CMM, the organisations with whom I have worked would primarily be at Level 1 (Initial) and Level 2 (Repeatable), approaching Level 3 (Defined) and in some areas Level 4 (Qualitatively Managed). Despite best intentions, none of the organisations could truly be said to have formally attained Level 5 (Optimising).

I have also been involved with Quality Management environments which have been informed by, and made at least some progress toward compliance with, ISO 9000.

Service Management

While implementing various elements of Service Management, most organisations with whom I have worked have not formally identified Service Management as a discipline. During my last few months with QPS, there was a movement to the adoption of the Information Technology Infrastructure Library (ITIL) method. Salmat/HPA is similarly moving toward adoption of ITIL, but to date this has had little direct effect on my activities.

Despite not formally identifying ITIL or similar in their approaches, all the organisations I have worked with have employed elements of those methods, particularly the creation of suites of formal procedures and policy frameworks for those procedures. Very frequently those procedures and policies have been oriented toward Change Management and service provision as defined by agreed Service Level Agreements (SLA).

I have been instrumental in the creation of procedures and policies throughout my career within State and Local Government, particularly as related to the production and maintenance of software and documentation. I have been subject to, and been involved with or responsible for creation and design of:

- database change procedures;
- system operation and management procedures;
- software change procedures;
- document management procedures;
- database and software design policies and standards;
- service level agreements.

Quality Management

I have been subject to formal Quality Management (QM) systems at both QPS and DNR, particularly in relation to the Polaris and IVASe projects. I have been an active promoter of Quality Management procedures, and have contributed to QM systems by creation of:

- documentation standards;
- programming standards;
- document templates;
- creation of the Polaris Code Review Process;
- creation of Polaris Code Design standards and guidelines.

I am particularly cognisant of the importance of QM systems in attaining compliance with Level 2 and Level 3 in the CMM. While State and Local Government IT expenditure is now strongly oriented to a project basis, the cost of maintenance of legacy systems and completed projects is high. Promoting organisations to higher levels of the CMM through introduction of sound QM practices can significantly reduce both project and maintenance costs by establishing unambiguous policies, guidelines and procedures.

Testing

Throughout my career as an IT professional, I have worked within the traditional regime of progressing changes through unit, integration, end-user acceptance and (where required) volume tests. In addition to this, in the last few years I have adopted and strongly promote a test-based approach to software design and construction.

For each employer where I fulfilled the role of Analyst/Programmer, I have had the responsibility for designing and running unit, integration, end-user acceptance and volume tests. This has been both for my own program code, and for programs and systems written by others. Wherever possible I prefer that programmers do not develop tests for their own code, other than unit and regression tests, but instead promote peer review and peer testing of program code. At a minimum I encourage peer review of test specifications and test results.

I have been responsible for the creation and application of vigorous regression tests, particularly in the context of introduction of new versions of RDBMS software, system management tools, and program libraries.

Through my exploration of Agile Methodologies, particularly XP, I now strongly advocate a test-based approach to software design and construction. If design-by-contract is used in specification of systems and system components, then

the unambiguous definition of the behaviour of the system or component in terms of pre- and post-conditions automatically specifies the tests that must be passed. When coupled with the Pragmatic Programming idea of writing just that code that is required to ensure that the tests are passed is applied, then the scene is set for rapid production of program code which has a very low defect rate. My experiences with these methodologies strongly support industry research in these areas.