Home My CV

CV: Mike Kinghan

I am looking for my next job!
If you're recruiting and you like my CV, email me at
imk@abstract-art.co.uk
Download a full PDF or a short PDF for hardcopy, or If you're reading the hardcopy, the webpage is
http://coan2.sourceforge.net/index.php?page=cv

Ideal jobs?

Development, integration, or build & test tooling for an innovating software enterprise.

Synopsis

I am a widely experienced software engineering team leader and software engineer, ScrumMaster, Previsor/Brainbench certified Master in C++ and C, open-source developer of Coan, "The C Preprocessor Chainsaw".

My skillset includes advanced C++ and C; Perl, Python and Bash scripting; GNU toolchain/Make/autotools; cross-platform Windows/Linux development and porting; low-level compiler/library/linker adaptation; test, integration and codeline management techniques.

I am equally fluent in Windows and Linux development environments. My deepest professional experience is in mobile operating system and SDK integration, build and test. I have worked in other sectors including telecomms, enterprise financial software, classroom systems, market research systems and CRM systems.

I have a Previsor/Brainbench rank in the 99th percentile for C++ proficiency and in the 100th percentile for C.

CV: Headlines

I have done a lot of things in the software industry. These are the last decade's *headlines. "More..." links will take you to the details. And if you care, there's* Ancient History.

Integration & Test Consultant, Ideaworkds3D, London: Sept 2011 - Dec 2011

- Introducing mature processes and practices for codeline integration and testing in a small, fast growing development business making the Marmalade SDK. More...

Tags: C++ • Test Driven Development • mobile • Agile • Scrum • test • test automation • test tools • continuous integration • unit testing • system test • integration testing • static analysis • cycloamatic complexity • code churn • quality metrics • quality assurance • SCM • codeline management • branching policy • process definition • Marmalade SDK •

Break: Dec 2010 - Sept 2011

- Made redundant in Jan. 2011 with the demise of the Symbian Foundation and took a gap-spell studying and developing Coan, the open-source project whose website you are viewing now. Took Brainbench Master's certfications in C++ and C. You can see my results at Brainbench by entering transcript number 5642877 in the box labelled "View public transcript." Learned a lot about the new C++11 Standard for C++ and applied it in coan 5.0.

Test Lead, Symbian Foundation, London: April 2009 - Dec 2010

- Symbian is an open-source operating system that powers mobile phones and dominated the smartphone market until it was swept aside by Android in H2 2010. I led the small team that integrated, developed and operated the Symbian Foundation's testing and Quality Assurance capabilities, supporting and supported by the Symbian community. More...
- Symbian encouraged everybody to get involved in initiatives where they had relevant skills, not necessarily confined to their official job. In my last 6 months at the Foundation I did a lot of kernel-hacking and bug-fixing for the GCC Surge Initiative, a push to get the core Symbian platform building with an open-source ARM GCC toolchain. More...
- I also took the opportunity of the GCC Surge to port a servicable subset of the Symbian build tools from Windows to Linux and also the Symbian QEMU virtual machine manager, making it possible for the first time to build the Symbian OS kernel on Linux and boot a Symbian VM on Linux. More...

Tags: Team leader • Symbian • operating system • mobile • embedded • open source • Agile • Scrum • test • test automation • build tools • test tools • on-target testing • binary compatability • GCC • C++ • C • ARM assembler • toolchain internals • porting • kernel-hacking • Linux • Ubuntu • Mercurial SCM

Technical Team Leader, Developer Build & Test, Nokia, London: Jan 2008 - April 2009

I was the technical leader of the team that developed, maintained and ran the automated build and test service for Nokia's Symbian OS development teams globally. More...

Tags: Team leader • Symbian • Nokia • operating system • mobile • embedded • automated build & test • build tools • test tools • on-target testing • Perl • C • Visual Studio • shell-scripting; • Windows • RedHat Linux • helpdesk

Technical Team Leader, Developer Build & Test, Symbian Software Ltd, London: May 2005 - Jan 2008

The same job as the one above, before Nokia's acquisition of Symbian Software Ltd. More...

Integration, Engineer, Symbian Software Ltd., London: August 2002 - May 2005

"Founder member" of the Integration team in Symbian Software Ltd., putting the pieces together to make Symbian OS. More...

Tags: Symbian • operating system • mobile • embedded • Platform integration • integration engineer • master codeline management • source control management • codeline quality control • build failure analysis • Perforce SCM

Consultant Porting Engineer, Systems Union, Farnborough UK: June 2001 - July 2002

Contracted to port Systems Union's enterprise finanical management solution, SunSystems, from MS Windows to Unix. More...

Tags: Cross-platform development • Windows • Unix • Linux • porting • platform abstraction layer • C++ • Visual Studio • GCC • Standard Template Library • POSIX

Consultant Unix Software Engineer, BT, Bracknell UK: January 2001 - June 2001

Contracted to rewrite the Solaris server daemon that supported BT's Directory Enquiries service. More...

Tags: Unix server development • Solaris • multi-threaded development • speed-optimisation • C++ • Visual Studio • Sun Visual Workshop • Standard Template Library

Consultant Windows System Engineer, Research Machines PLC, Oxfordshire UK: June 2000 - December 2000

Contracted to implement security customisations for the company's classroom workstation system. More...

Tags: Windows services development • classroom systems development • Windows security engineering C++ • Visual Studio • Standard Template Library

CV: Recent History

More: Integration & Test Consultant, Ideaworkds3D, London: Sept 2011 - Dec 2011

Ideaworks 3D Ltd. is a small fast-growing business that develops and markets the Marmalade C++ SDK for cross-platform mobile application development, targetting the Android, iOS, BBX, Symbian and Bada mobile devices OSes as well as Windows, Mac OS and Linux.

The ambitious growth plans of the business called for the SDK development operation to scale up dramatically within a year across several sites, but the existing development culture was that of a single small co-located team. The business needed to learn know how large-scale distributed software development can be done to demanding standards of speed and quality and its delivery on both counts was in jeopardy. It was just beginning to adopt Agile development methods. Quality assurance was an under-resourced and under-skilled activity performed post-development, routinely curtailed under release deadline pressure. The high-defect rate in major releases was "compensated for" by a hectic cycle of bug-fix releases.

I was hired as a consultant on the basis of my experience and record in the delivery of Symbian OS to introduce knowledge of mature techniques of SDK integration and testing that can realize rapid, high-quality delivery of "big software", to define the processes involved and create a 9-month plan for putting them into effect. The proposals I developed included:

- Introduction of metrics and associated KPIs for:
    - Defect backlog volumes, defect introduction rate, defect cleanup rate, and introduction/cleanup ratio.
    - SDK build & test success rate in continuous integration.
    - Test coverage.
    - Code-churn and code-churn/Agile story-point ratio.
    - McCabe code complexity.
- A Test-Driven Development Model to make quality assurance a controller of the whole development cycle.
- Creation of a new Integration function with responsibility for SDK mainline integration management and the defense of mainline quality.
- Introduction of SCM branching techniques and policies to enable successful scaling of distributed development, disperse defect risks away from the mainline and cut the high incidence of mainline breakage and regression.
- Introduction of static analysis tooling to automate detection of functional defects.
- Introduction of a standardized release test plan, based on a requirements-driven system test suite and a reference matrix of supported devices and OS versions.
- Recruitment of appropriately skilled test and integration engineers to the same grades as developers and in adequate ratios to development engineers.
- Migration of the continuous integration system from "underfoot" office-based kit to a managed virtualized infrastructure providing much greater reliablity, flexibility and scalability.

In the course of my 3 month contract I participated in the Engineering leadership team to initiate the execution of these proposals and recruit the new test and integration engineers. A permanent position was on offer to continue leading Integration & Test for the business, but though initially enthusiastic for this offer I ultimately declined it on account the extreme commuting entailed by the role.

More: Test Lead, Symbian Foundation, London: April 2009 - Dec 2010

Technology teams at Symbian Foundation worked the Agile Scrum way. I was the

ScrumMaster of the test team. I was also the Product Owner for the test team. (Product Owner and ScrumMaster being the same person is flagrantly against the rules of Scrum, but we didn't have enough people in Test to have one of each.) As the Product Owner, I decided what concrete objectives the team worked to deliver. As the ScrumMaster, I facilitated the team in achieving its objectives through each sprint.

One "half" of the test team's job was to test daily Symbian platform builds to gauge progress, or regression, against plan, and to test Symbian development kits for serviceable quality. The other and equally important half was to make the Symbian platform and its components suitably testable for all its users and contributors - OEMs, operators, platform and application developers, enthusiasts, and the Symbian Foundation itself. Some constant themes of the test team's work:

- Building our capability to automate testing instead of doing it by hand.
- Cutting out reliance on proprietary solutions; bringing up open-source ones.
- Bringing target hardware (or virtual hardware) devices on-stream for testing.
- Building and mantaining our capability on a range of key testing fronts:
  - Smoke-testing the platform builds.
  - Regression testing the upstream packages.
  - Key Use-Case testing of Symbian development kits.
  - Binary Compatability testing of new Symbian releases with prior releases
- Running test operations and publishing the results.
- Maintaining the test-related pages on the Symbian Developer Wiki.
- Connecting with the Symbian community and ecosystem to support their testing activites and foster their contributions. The *Symbian Test Working Group*, the *Build, Integration and Testing forum* and the *Bug Squad* were key channels for community engagement.

More: GCC Surge, Symbian: June 2010 - December 2010

The Symbian Foundation open-sourced the entire Symbian code-base under the Eclipse Public License on 4th Feb. 2010. It was the largest open-source migration ever, but its lustre was tarnished by the fact that you could not build the platform - or even the Symbian kernel - with open-source tools. Developers still needed to licence ARM's closed-source and costly RVDS toolchain.

The GCC Surge was a volunteer initiative launched by Symbian in June 2010 to port the core platform to CodeSourcery's open-source ARM GCC toolchain. I strongly supported the open-source motivation of the project and I had good enough skills in C++, C, ARM assembler and toolchain internals to take on the porting of the kernel package.

The Surge team succeeded in building basic Symbian Syborg ROMs with the ARM GCC toolchain and running them successfully in a QEMU virtual machine. The code was committed to the Symbian developer repo and I published *Howto build and run a Symbian QEMU ROM with GNU/Linux_tools* for the Symbian developer Wiki.

We proposed fixes for 86 portability bugs to upstream package owners in Nokia, of which 46 were proposed by me.

More: Porting the Symbian build tools to Linux: June 2010 - December 2010

Historically the Symbian build and development tools ran on Windows only. Supporting Linux hosted development had been a back-burner objective of the Symbian Foundation since inception. I felt that to push the Linux porting of the tools was a natural and necessary corollary of the GCC Surge if its play for open-source respect was to be forceful.

While my colleagues in the Surge did their work in Windows, I did mine in Ubuntu Linux. I built all the Symbian tools I was using on Ubuntu with the native GNU toolchain and applied Linux-portability fixes to them as need arose. This produced a Linux-hosted subset of the Symbian build tools that was equal to building the kernel, Symbian QEMU, the Syborg baseport and successfully running basic ROMs. These tools were committed to the Symbian developer repo and I published *Bootstrapping the Symbian build tools on Ubuntu Linux* for the developer Wiki

More: Technical Team Leader, Developer Build & Test, Nokia, London: Jan 2008 - April 2009

After Nokia's acquisition of Symbian Software Ltd. (not to be confused with the Symbian Foundation) in January 2008 I continued as technical leader of the Developer Build & Test System (DABS). DABS provided Nokia's Symbian OS development teams in UK, India and China with an automated service to build, test and metric their pending commits to the master codeline, and to publish the results on a central webserver. It was the principal quality-gate in the development cycle. DABS clients ran on distributed pools of dedicated Windows machines accross the development sites, pulling the code to build from a global Perforce repository and uploading results for publication to a RedHat Enterprise server. DABS was developed wholly in-house in Perl, PHP, XML, C and batch/shell scripts.

My team developed and maintained DABS, sysadmin-ed the system and manned the DABS helphesk. In 2007, an enlightened turn of company strategy pumped money and resources into building up and modernising the DABS hardware infrastructure - previously a shoestring operation. In tandem with this build up my team radically overhauled the system software to scale with the hardware at its disposal. When I transferred out of the job to the Symbian Foundation in April 2009, the capacity of the system had grown from ~1500 component builds per month to ~40000.

More: Technical Team Leader, Developer Build & Test, Symbian Software Ltd., London: May 2005 - Jan 2008

Prior to 2005, Symbian Software Ltd. had no organised build & test service to support development teams. Each development team was responsible for provisioning its own build & test kit, developing its own "nightly" build & test system and running it. The build & test results of one team were not routinely available to others who needed to know them. The build & test system owned by the London Networking team - called "the Steve Butler Build System" - emerged as a Darwinian winner and by 2005 was being used and developed on a skunk-works basis by about half the development teams in the company.

The spread of the Steve Butler Build System and its consumption of developer effort made the strategic importance of the function it fulfilled apparent to the software engineering leadership. A decision was taken to formalise the function within Product Delivery operations, where I was at that time the senior integration engineer. A small team was formed under my leadership to take ownership of the Steve Butler Build System, develop it to fitness for all teams' needs and global deployment, maintain it as such and run it as a developer service. The acronym DABS was chosen at this time.

More: Integration, Engineer, Symbian Software Ltd., London: August 2002 - May 2005

I was the first integration engineer hired by Symbian Software Ltd. Until 2002, the company had no organised function to integrate releases of Symbian OS. Master Codeline commits were made by the release engineers of individual development teams and the Master Codeline was thereafter the responsibility of the system build, system test and release functions. The escalating scale, complexity and pace of the development effort foregrounded the need for a dedicated integration function mediating between release planning, development and system build, and controlling Master Codeline commits. I was joined within a year by 3 more integration engineers and an integration manager to make up the team.

The mission of the integration team was to expedite the inflow of Master Codeline commits and raise their build & test success rate to a level that enabled the business to ship a release of Symbian OS to licencees every two weeks. This was a green-field challenge for which the team evolved answers that defined the integration function.

- We implemented and operated "air traffic control" for Master Codeline commits "piloted" by release engineers, so that they came in the right order with respect to their interdependencies and in good time with respect to release feature shedules.
- We implemented and operated a quality-control checklist for Master Codeline commits to head-off the common causes of system build or system test failures.
- On a rolling basis we projected required Master codeline commits from release plan status, lobbied release engineers for the commits required from them, fed back the commit status to release planning.
- We diagnosed system build and test failures on a daily early-morning basis, raised defects against the teams responsible and pursued commits of remedial fixes in time for the next day's system build.
- We monitored the comparative success rates of commits per development team, identified patterns of failure and engaged poorly performing teams in remedial action.
- We continuously worked on developments to automate and speed up the air-traffic and quality-control elements of our function.

We used automated "traffic-light" criteria to classify nightly system builds: red = broken; amber = serviceable for further development; green = clean. We measured our own success by the percentage of green/amber system builds. When the integration team was formed green or amber system builds were running in the low 30%s. Within a year this had been improved to the mid 60%s. When I moved on from the role in 2005, our skills and our automated processes were slick; green/amber system builds were expected to run above 90% and consistently did.

More: Consultant Porting Engineer, Systems Union, Farnborough UK:
June 2001 - July 2002

Systems Union's flagship product, SunSystems, had been developed exclusively for MS DOS and latterly Windows using Microsoft and Microfocus Cobol technologies for more than 10 years. It had a 3-tier client server architecture with a Microfocus Cobol backend. The client and the middle-tier server are written in MS Visual C++. Obsolete Rogue Wave class libraries and Apache Group's Xerces XML library, built from source code with in-house patches, were key load-bearers in the 300,000+ lines of server code.

The company has a strategic goal of reaching the enterprise Unix market, but porting efforts going back to 1999 had come to grief. I was hired to produce a porting strategy for the middle-tier server and to take the lead in its implementation. The target Unix platforms were Solaris, AIX, HP-UX, Tru64, Unixware and Linux, with Linux serving as the "guineapig" target, because Linux was free. For a mix of commercial and engineering reasons the company ruled out the obvious answer of purchasing an off-the-shelf porting solution such as Mainsoft's Mainwin.

In these circumstances I proposed a Platform Abstraction Layer (PAL), i.e. an API that would shoulder the server's platform dependencies, replacing all the Windows libraries. The porting problem would then be reduced to implementing the PAL on Windows and on each of the target Unixes. The PAL API would be demand-led. It would emulate just as much operating system as the server needed or looked likely to need. It would be POSIX compliant as far as reasonably possible to minimise diversity among the target implementations, and provide a portable system registry similar to the Windows registry. To maximise the uniformity of the build environment across all platforms I elected to use the GNU toolchain and the STLPort Standard C++ library across the board (with the necessary exception that the Visual C++ would be retained on Windows).

I won acceptance for this strategy in August 2001. One further contractor was then hired and the implementation occupied us until June 2002. The PAL reached a critical maturity in January 2002, when we were able to demo the server, built on the PAL, running on Windows NT, Windows 2000 and RedHat Linux 7.1. The PAL then comprised about 20,000 lines of C++.

The planned fan-out from Linux to the proprietary unixes was interrupted. In May amid corporate financial retrenchment the Unix port was sacrificed for the current financial year and my engagement concluded with mothballing the project for resumption in more prosperous times. In 2003 I learned that it had been resumed, with the fan-out development out-sourced in China.

More: Consultant Unix Software Engineer, BT, Bracknell UK: January 2001 - June 2001

I was hired by BT (British Telecom) to rewrite a Unix server daemon that supported the Directory Enquiries service for BT and other telecos. The re-write was called for by the planned transition from a national fleet of 44 Sun Solaris servers, each running an instance of this daemon in support of a single Directory Enquiry Centre (DAC), to just one Sun server (plus a backup) running a single instance of the daemon supporting all DACs nationally. This single instance would therefore experience about 40 times the previous demand, and its efficiency needed to be made adequate to this multiplied workload.

The server was targeted on Sun Solaris 2.6 and 8, and to be written with Sun Visual Workshop 5.0. I was able to achieve the necessary efficiency gains by:

- Redesigning the server's call-cache data structure for much faster access, at the expense of memory efficiency - since memory was not a limitation.
- Redesigning the server architecture to be a fast multi-threaded one instead of a traditional but slow forking one.

When I had successfully demoed the new server in June 2001, my engagement on the project was concluded and the national roll-out was undertaken with in-house resources.

More: Consultant Windows System Engineer, Research Machines PLC, Oxfordshire UK: June 2000 - December 2000

Research Machines (now RM) is a vertical IT solution provider in the education sector, with a powerul position in the UK. The company offers specialised classroom systems, from hardware and networking infrastructure right through to management software and student applications, built on Microsoft OSes.

I was hired to research, specify, and engineer system security customisations of the company's first Windows 2000-based classroom system. Classroom network systems call for security hardening much beyond the levels usual in business environments because their legitimate users subject them to far greater levels of ignorant abuse, hacking, vandalism and surreptitious software installation than are found amoung business users, and because the tech support resources available for system maintenance are very meagre by business standards.

I concluded this project in a much shorter time than scheduled by discovering a third party product from Appsense Technologies (a specialist in NT security solutions) that I could integrate, rather than writing the solution myself. The project manager was wholly satisfied with the Appense product and rapidly concluded a licensing deal.

So for the remainder of my contract I undertook other developments for the classroom system. These related to enhanced user-management features and BIOS security. Since many components of the system were to be NT services, my most valuable contribution was probably the development of a C++ template class that implemented all the boilerplate functionality of a Windows NT/2000 service. This template class enabled RM programmers without prior experience of NT services to develop them very rapidly by specialising and subclassing my template.

CV: Ancient History

IT Technical Consultant, ACNielsen Ltd., Oxford, UK: 1993 - 2000

ACNielsen is the world's largest market research company. I worked at the Oxford site, both as a contractor and an employee, over 7 years on enterprise development projects including:

- Development of handheld barcode scanning applications, in MS C, dBase and FoxPro, for in-store product data capture by ACNielsen field auditors
- Development of a ACNielsen's first wide-area electronic data-delivery service for transmission of market-research databases to the company's customers. The service, called ACNielsen Connect, replaced courier deliveries on CD. It was written in Visual C++, with Unix and MVS mainframe backend interfaces.
- Development of a middleware service, ACNielsen Director, for large customers to automate aggregation of market-research databases received on the customer's ACNielsen Connect clients into the customer's Unix or MVS data-centre.

Independant software developer and consultant trading as Turing Tools, Oxford, UK: 1992 - 1996

I developed and marketed security customisation products and programming tools for the Acorn RISC OS market, selling mainly to education, local government and small businesses. Products were written in Norcroft ARM C, ARM assembler and BBC ARM Basic. The terminal decline of the Acorn ecosystem after 1995 made this business unviable and I migrated into Windows development consultancy.

Senior Systems Analyst, Principal Programmer, AT&T Istel Ltd., Oxford, UK: 1988 - 1992

AT&T Istel at this time supplied all the computing and systems development needs of the Rover Group, PLC. I worked in an IBM mainframe and PC environment, mainly on Rover's customer-dealer-vehicle PL/I, DB/2 database systems and on the warranty and marketing systems which these databases fed. Amid large-scale layoffs in 1992 I accepted voluntary redundancy and started my RISC OS software development business, Turing Tools

Systems Analyst, Lazerlink, Oxford, UK: 1986 - 1988

Lazerlink was a direct-marketing systems spin-off of the Rover Cars Group. My first job in IT, a baptism of fire in an intensely pressurised MVS mainframe development shop.

CV: Education

My education had nothing to do with software development. I have a 1st class degree from Queen's University in my native Belfast, in Philosophy, History & Philosophy of Science.