The Past and Future of Australian Innovations in Information and Communication Technology (ICT)

Oral History Interview

40

Jeff Whittle

Interviewed by:

Sebastian Boell

Interviewed on:

Monday 20 June 2022

Citation: Boell, S.K.; Philipson, G.; Thorne, P.; Riemer, K.; Peter, S.; Wang, B. (2023). Series of Oral Histories with Pearcey Medallists and Hall of Fame 2000-2020. Sydney, Melbourne. DOI: doi.org/10.25910/j5ak-rs49

Project Summary

This interview is part of a series of oral history interviews undertaken by the Pearcey Foundation and the University of Sydney as part of the project 'The Past and Future of Australian Innovations in Information and Communication Technology (ICT)'. The series interviewed recipients admitted into the Pearcey Hall of Fame from 2003 to 2020. The hall of fame recognizes outstanding life-time contribution to ICT in Australia in business, research, and government. Each oral history captures a short biography of individuals who made an outstanding contribution to ICT in Australia. They also collect insights on aspects that had a lasting effect on ICT innovations in Australia, positive as well as negative from approximately the 1960s to the 2010s. Interviews lasted about 60-90 minutes and were conducted by Sebastian Boell, Graeme Philipson, Peter Thorne, Kai Riemer, Sandra Peter and Belinda Wang. The complete set of interviews in this series is archived by the Pearcey Foundation.

Key Points Covered in this Oral History

1. One of the factors that led to Whittle programming's success is the close location to customers and the market (WA mining). They also ran a series of conferences back then to showcase the use of their products in mining, which were well welcomed by the local mining industry people.

2. The government support company's research and development in terms of tax incentives. It is said more helpful when assisting local businesses offshore, which (Austrade) provides invaluable support when selling the product overseas.

3. The conservative culture has slowed down IT innovation locally, but the inclusive culture makes the local business a fair competition regardless of their backgrounds which might promote entrepreneurs and innovations.

Biography

Jeff Whittle AO

Founder of Whittle Programming
Fellow of the Australian Computer Society
Pearcey Hall of Fame in 2020

Creator of Whittle mining software series, Entrepreneur

Jeff Whittle was born in England. He trained as physicist and worked at the Medical Research Radiobiology Research Unit at the Harwell Atomic Energy Research Establishment. He and his family moved to Australia in 1961, where he joined the newly created Computer Centre at Monash University.

In the early 1980s Jeff turned his attention to developing software which would optimise mining operations. So successful has he been that Whittle software is considered to be an industry standard within the international mining community.

It is said that mining engineers, around the world, know that their pits are not optimally designed until they have been 'Whittled'.

He has been the pioneer in the application of techniques such as pit optimisation, mine design optimisation, scheduling, cut-off and strategic mine planning.

In 1983 Jeff Whittle wrote Three-D. The development of this software is said to be one of the most significant software developments to optimise mine design it was the first implementation of a pit optimisation algorithm that could be used with any mining software package on any computer. This later became Four-X, when the ability to handle multiple elements was added. Four-X has become an industry standard for open pit design. It is used by many mining companies world-wide. The Four-X technology was sold to Canadian mining software house Gemcom in 2002.

From 1998 onwards Whittle's interest had been in optimising the schedules of very large mining complexes, leading to the development of Four-D, where the fourth dimension was time. His software can produce optimal, or near-optimal, long-term schedules for complexes with hundreds of pits, multiple processing options, multiple products that have to be blended, and multiple stockpiles. His more recent Prober software has become a trusted benchmark for Enterprise Optimisation in the mining industry. This has delivered lasting and quantifiable economic, environmental, commercial and societal benefits.

Jeff Whittle was the originator of what is now the GEOVIA software. Currently, there are over 2,000 Geovia Whittle software licenses in use by mining, exploration, consultant and academic customers around the world

In September 2006 the Council of the Australian Computer Society elected Jeff Whittle to the grade of Fellow.

Jeff Whittle was recognised in the Australia Day Honours for 2018. Jeff has been named an Officer (AO) in the General Division of the Order of Australia for:

> "Distinguished service to the information technology sector and the mining industry, particularly through the development of industry standard computer programs for strategic mine planning and optimisation."

Interview Transcript

Date of interview: Monday 20 June 2022

Can you briefly summarise your biography, where did you grow up, what's your education, and what were the names and places of some of your organisations? Normally this question is taking most of the interview.

No, it won't take that long, but I grew up in Lancashire in England, in a little town called Aspull and I went to school there, of course, and I got a scholarship to go to Manchester University, and I went there for three years and got an honours degree in physics. People query how you can get an honours degree in three years, but in those days, they played it tough, it was a very strict system for getting into the course and if you failed a single exam, you were out. So, our year started with 70 students and ended up with 40, and even not all the 40 got a degree at the end, so it was a different approach from current day where you can take a supplementary and this, that and the other. No supplementary, it was black and white, so that's how I got an honours degree, just a second-class honours degree, I didn't get anything higher. And basically, most of my training was in experimental physics, my skill is in making things work, which actually applies to computing just as well as it does to physical objects. After that it was a couple of years doing my National Service, and then I very casually applied for a job at Harwell, the Atomic Energy Research Establishment at Harwell, and I didn't take too much notice with the job they offered me, but it turned out to be an assistant experimental officer. Generally, you have the job of sweeping the floors and things like that, you were very low class, and I was young, I didn't care. It was a while before I found that that actually completely denied me getting into any serious research, so it took a while to get over that barrier. In the first few years there, I lived at what was called Ridgeway House. Harwell was built on a wartime airfield, and it had of course an officer's mess and a sergeant's mess, and the officer's mess became a staff club under Harwell. In the hall of residence there were 70 men and seven women, and I got lucky, I married one of them, since that we've had six children and 12 grandchildren, and that's over the years. When we got to sort of four and a half children we describe it, which is 1961, we were right in the middle of the Cold War in Europe, and while we never thought there was going to be a nuclear war, it was certainly a possibility of a nuclear war. If there was a nuclear war England was no place to be, it's very small, the size of Victoria. Khrushchev had promised England 18 hydrogen bombs which meant there was going to be no part of England that was worth living in, and we had four small children and another one on the way. And you think differently when you've got children, as I'm sure you know, and so one night we decided to get out. What wasn't commonly known at the time, and even now, was that the atmospheres in the northern and southern hemispheres don't mix much, and at that time any nuclear war was going to take place in the northern hemisphere, ergo, you go to the southern hemisphere. And if you do that and you want to speak English, at that time it was Australia, New Zealand, or South Africa. We didn't fancy South Africa. I had an uncle and aunt in Melbourne, who I'd never

met because they'd come out to Australia before I was born, but at least it was a point of contact. So, one Monday night we decided to come, two and a half weeks later my wife flew with four children and one on the way, and I followed a month later, having packed up the house. You can't pack up a house with four small kids in it. Not feasible.

Yes, I think that's a very important point because I was not sure why you came to Australia, but can I go back a little bit, you said you were starting to program in 1962…?
Yes, that was in Australia.

How did you come in contact with computers, when was the first time you used a computer?
All right, April 1962, so I've been programming for just over 60 years now, though I don't do a lot these days. What happened was I got a job at Maribyrnong at :The Department of Supply as a physicist, and within two or three months I found that I was going to have to do a calculation as part of my physics that was going to take me four months sitting at a desk calculating. Now, a couple of things wrong with that, one I'd be stark raving mad at the end of it, and the second there's no way you can get the calculation right, you cannot work for four months and not make a mistake. And just at that time the first bureau computer became available in Melbourne, in Kings Road Melbourne, Ferranti Sirius, discrete components and all that sort of thing, we thought it was fantastic, it did 30 instructions a second. That was marvellous. Anyway, I went and learned to program that, and I got my first program right first time, and I thought oh, maybe this is something I'm good at. And over the next six years, still working for the government, in Maribyrnong, I did more and more computing and less and less physics. And at the end of that time I moved to Monash University in the emerging Computer Centre there. And I ran/started the Computing Contracts Group, where we did work for outside organisations to raise money to buy equipment for the Computer Centre. Mainly we did examination work for various states, mainly Victoria but sometimes New South Wales as well and other states. It started with myself and a girl part-time and we ended up with a dozen staff, and I was there ten years. But, of course, half the year you're not working on exams, so we played. I was very fortunate there were some really seriously bright young fellows, there (there was one girl, let's not be too sexist) and I learned an enormous amount in that ten years. There are techniques I learned in computing that I still use and so I regarded that as a huge learning thing. At the end of the ten years, I got a job with Newmont Mining Corporation, this is how I got into mining, and I was…

Yeah, so was it part of your orientation or your job that you tried to use the computer at Monash for multiple purposes and to get external funding, or was that your initiative?
That was already in train, getting contracts, what we didn't have was someone to actually do them, so they hired me. I don't know if it still goes on, nowadays of course, they've got a Computer Centre and the Computing Department, they're two separate things, we were only the one in those days, very early days. The Ferranti Sirius, 30 instructions a second, we thought that was fantastic.

Was that part of the reason you got the job is because you were, no, sorry, in the UK you didn't work with a Ferranti Sirius, or you did as well?
No, I didn't touch computers in the UK.

Okay, now I get it, yeah.

Yeah, so it all started with this calculation, a four-month calculation, that I had to find a different way to do. The calculation eventually took about two hours, but it took me a month or two to write the program, so it was not my first program, it was a proper program.

You were faster than the four months you would have taken otherwise.

Oh yes, yeah, absolutely, but it was interesting, and I started to use computing in the physics more and more and gradually. I now call myself a 'computerist' rather than a physicist, but I still think like a physicist.

And what triggered your move from, or maybe how would you describe how a physicist thinks?

Oh wow, no-one's ever asked me that before. I think I'm always figuring out how things work, whenever I see something, I want to know how it works. When we were first married, I used to drive my wife mad because everything we bought, I took to bits, but I always could get it back together again, but I was interested in how it worked. So, I'm quite uncomfortable if there's a physical thing happening that I don't understand. I want to know, that's I guess not just a physicist, it's a scientist. But on the other hand, I am not an academic, I don't think like an academic, I don't do literature searches before I start programming, which will show in the work I've done. But it's not that I disrespect academia, I think it's very important, and indeed, were it not for the academic paper by Lerchs and Grossmann, we wouldn't have a business, and that's how, for six or seven years, I was in effect Newmont's technical computing department, not their business computing department, different thing. And they hired me with the idea that they had an old IBM, and they were transferring onto a new machine, and of course the language changed. That was in FORTRAN, a sort of FORTRAN, a weird sort of FORTRAN, and they had a quarter of a million lines of code that they wanted moved across. They had the concept that they'd hire someone and he'd sit and read that line of code and write that line of code, a quarter of a million lines, but of course that's nonsense. But by then with my experience at Monash, where I'd worked on compilers and all sorts of things in the off times, and worked with people who were inventing in that area, I had a fair idea of how to parse text. And so. I persuaded Newmont that they didn't want to hire a programmer, they wanted a consultant who would write a translator for them, to translate from one machine to the other. And they took me on at $25 an hour, which was a lot of money in those days. I spent, I guess, about two months writing a translator, and it didn't translate everything, but it flagged what it didn't translate. Flagged what it didn't translate so you could, about 1 percent you had to look up by hand, but then I set about translating a quarter of a million lines of code, and that took about six weeks.

And were you aware of other people doing similar things or was this…?

No, no, I'm just doing it myself.

So that was really an uncommon approach that you would translate one computer language into another…?

As far as I was concerned, yeah. One of the curiosities about FORTRAN was if you had a two-dimensional array it was referenced by the second dimension, and that took some fiddling to correct it in the new program. So, once I'd finished that translation, by then I was indispensable at Newmont, I made damn sure I was, and I stayed on with them for quite a

while. Going on now, do you want me to move onto Whittle Programming and all that sort of stuff?

Yeah, so you didn't explain what the program was doing with a quarter of a million lines of code.

Oh sorry, yes, they were a suite of programs, not just one program, but I think a couple of hundred programs. Mining is very complicated, mining is a messy business, and you need all sorts of tools to start from the drill-hole information to create images so you can look at what the ore body is like. So all their programs were to do that sort of thing, to manipulate the data. In those days we didn't build it all into one package, which you can do now, you just ran a program to do this and ran another program to do this and so on, so those were the programs I was translating.

More like the UNIX philosophy, one program for one purpose.

Yes, yes, and to a certain extent I followed that. What happened was, earlier on I knew there was an algorithm written by Lerchs and Grossmann at the IBM Research Labs, which would optimally, mathematically optimally, find the shape of an open pit that maximised the cash flow. That's a surprisingly difficult thing to do. Starting from a block model, so it's regular blocks with an estimate of grade in each block, and you might have a quarter of a million of them and remembering that the slopes can vary in different places and in different directions. Working out the shape was very difficult, and it was well known in the mining industry that it had only been programmed by one company, a Canadian mining company, the paper was published in '65, and in the 70s someone got it working in Canada. And two things, one, everybody knew that it had cost $250,000 US to get it working, the other was it wasn't easy, and the other was, of course, only that company could use it. I realised that maybe the mining industry, as a whole, needed that program, and I read that a paper written in 1980 (I think I read it in about 1982) that said the mining industry would love to use this algorithm, but it's too hard to program. Now I had 20 years programming experience under my belt by then, and there was no way I was going to accept that statement, so I figured out how to write it and I did some tricks with it. I had to alter the algorithm as described, because with the computers then available you couldn't do it, the people in Canada had built it on a mainframe. I figured out how to do it, so I offered to write it for Newmont. "No, no, we don't want to take the financial risk,"- wouldn't have a bar of it. So we only had one child at home by then so I went to my wife and said: "Well, should we start this as a private venture, because the mining industry needs it". I thought, and it turned out it did. Now just backtracking, the mining shape, it's important to get it right, a mining company might spend $200 million digging a hole, they'd like to get that right, 1 percent is $2 million, that's a lot of money. Anyhow, my wife said: "Okay, well you've supported me in the raising of children, I'll support you with your brain children." And she took on the marketing, never having done any marketing before in her life, and over the next few years sold it in 35 countries, which was pretty fair at marketing, And without that, none of this would be. I wouldn't be here, because I couldn't market my way out of a paper bag, so that led to the formation of Whittle Programming.

So would you say then it is very important for a young computer company that you have technical skills but it's also important to have other skillsets, such as marketing or business…?

Oh, absolutely, oh yeah. I mean the old saying is if you invent a better mousetrap the world will beat a path to your door: no, they don't. It took us one year to make the first sale, I wrote it at home, I wrote it on a PC, very early PCs. We had to buy an IBM PC, which cost $13,000

then, and didn't have a hard drive. I eventually got a hard drive which held 20 MB. But anyhow, that was the sort of starting point of Whittle Programming. And I wrote what we now call Three-D, just the three-dimensional shape of the pit. Critically, it optimises cashflow, not discounted cashflow, that's a different problem, but it's a start, because mines operate over several years and the money you spend in the later years is cheaper than the money you've got today, because you could get interest on it. So that was the start of Whittle Programming. I went to a conference in the Pilbara and saw, and I can't remember the name of the person giving the talk, but they started to talk about parameterisation of pit shells, and I didn't like the method they were using, it was to my mind very messy. And I realised with Three-D that what you wanted to do was change, what I called the revenue factor, and as you increase the revenue factor the pit will get bigger. By definition it must get bigger, it can't get smaller at any point, and I realised that it was a good idea to have, not just one pit shell, but a whole series of pit shells that you could sort of use as guides for all sorts of things. And I figured a way of doing it that didn't need the computing effort required for 50 optimisations that was about two or three times what it was to do one pit optimisation, because I kept reusing information internally. And that led to some very simple scheduling. When I talk about shells it's the sequence in which you mine stuff, it's terribly important in mining, you basically want to mine high-value stuff early and low-value stuff later. That led to Four-D, which took into account time, over the next few years, then Four-X, which had more than one element. Up till then I'd only dealt with one element, gold or silver or whatever it was, and I'd written a program specifically for use by everybody. There were lots of mining packages around at that time which did a lot of these things that I'd been fiddling with at Newmont. But at that time they didn't optimise a pit, they could do all sorts of manipulations on it, but I wanted to write it in a way that any mining package could interface to, obviously to increase the size of the market. So, I kept it very simple, externally, it was complicated inside. It read a text file, thought a bit, wrote a text file, an ordinary text file. That's the easiest thing possible to interface to, that anybody can interface to and you define the format of the input file and anybody can interface to it, and that kept going for many, many years…

Can I go back a little bit? So you said it was computation, everybody in the industry is sort of like it's computationally too hard to implement this algorithm, except one company in Canada.

Yeah, so programming too hard, apparently, they had a hell of a time getting it working. I actually spoke, it was a woman who wrote it actually, I actually spoke to her on the phone once and we agreed that we wouldn't be swapping ideas. It was very good natured, yeah. The industry needed this or I thought they needed it. They didn't think they needed it, because this is another factor of mining, apart from being complicated it is very conservative, and there's a very good reason why it's conservative, and that is that every block, every cubic metre in the ground is different. Before they get there, they don't know in much detail what it's likely to be, so there's huge risks in mining. When people complain about miners making a big profit, they don't talk about the ones that failed, there's quite a high proportion, even today, of mines that failed because what they find down there isn't what they expected. So I understand that miners are conservative, so that's why it took a year to make our first sale. The first sale, I had to go back to consulting because we like to eat now and again, and in those days we were cheeky, we asked for the money up front and it was $5000 US. And one day in the mail arrived a bank cheque for $5000 US from a company in Spain called APIRSA, that we'd never heard of, how they'd learned about us we've no idea. So my wife and I packed up the software and took it to the Post Office, and went and had the most expensive lunch we've ever had before or since. So French champagne, the whole damn thing, caviar,

everything, why not? So that was the start but then sales built up and within, I think about six months, I was able to stop my consulting and work full-time on the company, and that's when I was developing all the other things that went with it, and that eventually got sold as a company. Whittle Programming got sold to a different company but in Canada, and in fact two of our staff and one of our sons went with it as part of the deal. So now I have grandchildren with Canadian accents, although they now live in Australia.

But I'd like to probe a little bit about the early days because I think they hold some valuable experience, so you were very confident that this would be a product that the market will respond to and your wife was supporting you with that, and how difficult was it to actually program it, did it take you as much time as you thought or…?
It took me a bit longer than I thought, but it didn't take that long. I do remember there was one particular spot in the code, two pages, about 50 lines of code that took me two weeks to debug. Remember, we didn't have all the tools we have today, none of those things, we didn't have a graphical screen or anything like that, so things were harder. But I fell into the trap that in FORTRAN in those days if you didn't specify what type of variable it was, if it had a name starting from I to N it was assumed to be an integer, you didn't have to specify it. And I got trapped by that because I mentally had a floating point value that I had given a name starting with an I. So that was two weeks to find.

And how long did it take in total to program?
I'm sorry, what?

How long did it take in total to program the first version?
The first version, I think it was only about six or eight weeks, something like that, yeah. I can't remember, to be honest. It wasn't that long because I had to go back to consulting because like I said, we like to eat now and again.

And did you then try to sell, you said like the first customer was a surprise, how did you try to market your product and why wasn't Newmont one of the first customers?
Oh, I realised that I needed real data for testing, real data is not that easy to get in the mining industry. So I struck a deal with Newmont that they could have a copy if they gave me access to their data for testing purposes, so that's why. It's funny, we used to go to mining exhibitions in the United States and in South America and we'd hire a booth, and I don't know if you know but they're huge exhibitions and conferences, they're 400, 800 booths, it's a big thing. And I do remember very distinctly in the early days with our booth we were selling Three-D, and I had this guy, mining engineer, standing in front of me and he said, "Now, my granddaddy showed me how to design a mine, that's good enough for me." That was what we were up against. So, the fact that he was wearing a Stetson made it funnier.

So you had already tried to sell your product, not just within Australia but right away overseas?
Yeah, right from the start, right from the start, we knew it had to be an international thing, we had to give it to every mining company.

And did you get support from the Australian government or…?
Yes, we did, we got some support for some of our expenses when marketing overseas, I can't remember the details now, but we did get that support and that was very, very valuable to us.

And more to the point, we would get consular support when we were overseas, and they would set up a section, in some cases, not all, in some cases, particularly in South America they tended to do it. They would have a group of booths that were all from Australian companies, and we would sort of work together in a sense, but then you have to speak Spanish. My wife learned Spanish just for the hell of it, yeah. I'm numerate, she's literate.

Was that with Austrade?
Yeah, Austrade, yeah. I believe they still do that sort of thing, but yeah.

Did you get any other kind of support for development or…?
No.

And how were you aware of Austrade, was that something you knew already or…?
I think my wife found it, I don't know, don't remember.

So she really drove the business side of things and gave you the space to concentrate on solving this back…
Oh yes, there was clear division. The other thing she did, she would say: "Look, they've got this competition, shall we enter that?" and I'd say: "No, no, that's not my type of thing," so she'd enter in any case, and we won quite a few of them. I was only told when we'd won.

But I think selling a product overseas was that tricky in terms of tax and other administrative problems that you would have had?
Yes, right in the beginning we sold source code, because not many mines had PCs in those days, the mid-80s, they all had mainframes, and every mining engineer learned FORTRAN. That was why I wrote the program in FORTRAN, because we sent source code to the mine with instructions and also some test data and all that sort of thing, and the mining engineer then had the job of installing it on their computer. Later on, when we were selling Four-D, which we were selling for $30,000 US, that price included me going anywhere in the world and installing it on their machine. And that quickly started to kill me for the travel, we had to hire someone to do it for us, who still works with us, still works with Consulting now. So around 2000 we tried to float Whittle Programming, well, we did float it but it wasn't a success, but soon after that ….. By then we had Whittle Programming and Whittle Consulting and Whittle Programming was sold to this Canadian company, Gemcom, but we retained Whittle Consulting. By then another of our sons, we've got five sons, and another of them who had very wide business experience in Europe and had been working in, you know DHL, the courier company.

Yeah.
Yeah, well he was Financial Controller for Europe and Africa, on a healthy salary, but he got sick of living in Europe and moved back to Australia. He saw what I'd done with the next step after Four-X, which. Four-D was a generalised optimisation program that optimised the whole thing, which we call Prober. It is a completely different technology internally. And he thought he could do something with that and indeed he has, so he runs Whittle Consulting now, and that changeover was about 2000.

Can you explain it a little bit or unpack it a little bit? So you said initially you were doing consulting work but it was general consulting, not in the mining sector, then you had Whittle Programming, you sold your first versions of your Three-D software…

Slow down a bit… when I worked at Monash I wasn't consulting, that was a salaried job, but when I moved to Newmont that was consulting if you like or contracting. It has various names but I was paid, that was initially for a specific translation job. But I did other things for them while I was at it and I became indispensable, and I stayed on there for quite a while. And that was what led me into mining. Mining is very interesting, it's very interesting. The program we now use in Whittle Consulting, Prober-E (we've had A, B, C, D, E), that has an input file with over 100 different types of data in it. So it's not an easy program to use, we don't sell it, you couldn't hand it to a user, because it's a year's learning to learn how to drive it. But we have some bright young fellows in the office (who we call "Tech Services") who do that, they know how to drive it. And they've actually got it to do things that I didn't plan for, but it survived, the mathematics was strong enough. So that complexity really, there's so many different aspects that can affect it, price, hardness of the rock, all sorts of different things from poisoning of one element with another and having to allow for that. It's very, very complicated, but we're in a situation now where we can tackle products with multiple mines, multiple processing plants, and optimise the whole thing as one object, and it's not a mathematical optimisation but it gets very, very close.

But the reason to get out of Whittle Programming and into Consulting, was that because the programs became too complex for…?

That was one of the reasons and probably the main reason. At that time certainly, we were the only company in the world, and to a certain extent we still are the only company in the world who could do that. It seemed to make sense to do it as consulting, because you can charge a lot for optimising the whole of the mining operation. This is background with Prober: Whittle Consulting has now been running for 20-odd years and over that time we've done what we call "enterprise optimisation", which is a big study. It starts from the customer's data, and in some cases their own schedule which they've often produced and which we call the base case, here's how they plan to mine it, and you can work out the value of it, what the net present value is just by looking over the cashflow. On average, we've done 160 of those studies. In about 100 of them there was a base case to measure against, and on average we increased the NPV by 50 percent. And that's serious money. And I don't believe there's anybody else in the market who can do that. So, we operate worldwide, and yeah.

At around 2000 when your son came back from Europe?

I'm sorry?

It started at around 2000?

Yes, yes, it did, yes, it started. He's certainly brought skills in again, ones that we didn't have. One of the mistakes that family companies make, over and over again, is they hire people like themselves. Right from the start we understood that when we hire people, we hire them different to bring in different skills, and Gerald, the son who is running it, had totally different skills from ours, background in accounting, things like that. He was Financial Director of an airline at one stage, the DHL airline, delivery airline. So, he had all sorts of business skills that we didn't have, so that was good.

Yeah, so one point I want to go back to Newmont, you said getting the data from them was crucial for you to really test your software.
Yes.

And how did this influence your ability to sell and market your software?
Well, we could quote Newmont's use of it, but that was about it. Incidentally, it took 165 hours the first run. Fortunately, I had built restarts into the program. I'm a cautious programmer, if it's going to take a long time I build in restarts, and we restarted many times in that 165 hours. But remember we had no plotters or anything, I don't think there were plotters, but we finally printed out an outline of the pit from their data and we did it with a line printer with different characters to show the different parts of it, and took it to the head mining engineer who was very sceptical about the whole thing. And he famously said: "Oh, that looks like a pit." We've quoted him so many times, he even quotes himself now. So yeah, it was doing the job properly.

So you say what was really helpful was being in Australia close to a mining company that actually helped?
Yes, yes. We ended up spending a lot of time in Western Australia, because that's where a lot of the mining is of course. And so at one stage we ran a series of conferences called "Optimising with Whittle" in Perth, and they were very popular, hundreds of people turned up for them. Basically, the conference is about using our software, so other things came in as well, and a couple of times, we set up a competition. We actually hired 30 PCs and put our software on all of them, in a room, and then the people from the conference, attendees, would enter the competition. They had standard data, and the competition was to get the best result, and boy was that competitive. There were mining engineers working… it was very competitive. They had a ball. We'd spend a day doing that.

And that would be the late 80s, early 90s?
Yes, yeah.

So then you decided to get out of the software business and into consulting?
Well, I was still writing, I still did the software, I wrote all the programs, not the data handling program for putting it into Prober, nor the output: that's tech services, that's their job, I'm just dealing with the engine. The engine is about 30,000 lines of code, and I wrote that.

So you must believe in the mythical man month then…
Yeah, it used to be $25 a line of code, fully developed code, I think, was quoted in the early days, well I certainly haven't cost that. Yeah, I wrote the original program in FORTRAN because of the history and I was using stuff from the other programs, some parts of it. But it became obvious that that wasn't a long-term solution because nobody teaches FORTRAN anymore. I did a survey of languages to see which other ones we should use, there was C++ and there was VB.NET and things like that, and of course you've got Visual Studio by this time, which I would have killed for in the early days. And I decided that VB.NET was the best language for us to do this because VB.NET was designed as a language. C++ has just grown like Topsy, it's a mess, it's a disgusting language. I know everybody uses it, people will swear, and I wouldn't try and write an operating system with anything else, of course you've got to have that sort of thing, but for applications it's a disgusting language, to my

mind. So, I said okay, well, we'll write in VB.NET, but by then another of our sons had joined the company to take over from me, and he flatly refused to work with VB.NET, he wanted C++. So I rewrote the whole thing in VB.NET, which took me a year, and I did it so that it produced bit for bit the same results, which took a lot of formatting and stuff like that. I couldn't use the standard formatting. And then I used a translator to translate into C++, which did most of it, you had to do a little bit of tweaking here and there. And again, I produced a C++ version that was bit for bit the same results of the other two, and by this time we've got a suite of over 200 test data sets that we run. Incidentally, with text input and text output, running a suite of regression tests is dead easy, all you've got to do is compare two text files, and there's plenty of software to do that, whereas when you're dealing with screens…... I couldn't write a payroll program to save myself, I wouldn't have the faintest idea where to start, so I'm strictly technical programming.

So you're still maintaining three different versions, one in FORTRAN?

No, no, they've gone. We've hired somebody else, a guy from Melbourne University, Peter Grossman, he's very good, and he's a mathematical academic but he's also fluent in C++ and so he is working alongside our son.

Yeah, and so can you talk a little bit about what role do you think it played for you? You make the move for different reasons from the UK to Australia, but to what extent was the Australia context really helpful or really influential to your success?

Absolutely very valued and influential, very strong. The best thing we ever did was move to Australia, for all sorts of different reasons. We frankly didn't expect to like it, that wasn't our reason for coming, our reason was to protect our kids, but we've loved every minute of it. And one of the things, you may not be able to hear it, but I have a north-of-England accent, and in the 50s and 60s that was a professional disadvantage in England, because if you've got it, if you're from Lancashire you must work in a mill or a mine, you can't have any brains. I mean my accent has faded a bit now but after all, we've been in Australia over 50 years now. But yes, I found Australians couldn't hear my north-of-England accent, didn't matter to them, and I was taken for what I could do rather than what I sounded like, and did very well.

So you think a part of the Australian context is that it really gave you a fair go?

Yes, yes, very much so.

And the context of mining, do you think looking back it was serendipity that got you into mining and that…?

Oh yes, yeah, just sheer luck, just sheer luck. The experience out at Monash, the ability to do parsing of text and how to do that and how to write compilers actually, I was involved in the compiler writing at Monash, and brought me into translating these quarter of a million lines of code, and that got me into mining, and I got very interested. I've always liked to try and do the optimal thing from a mathematical point of view, trying to optimise, in some form, I mean different things you want to optimise obviously. While I was at Newmont, even before I wrote Three-D, I worked on other optimisation things there, using Cplex in the very early days, which I'm sure you've heard of Cplex. I'm more than happy to use Cplex for linear programming. It's unbelievably fast and I used that, and that's used a lot in Prober, Prober uses a lot of linear programming, it runs hundreds of linear programs in a row, and some of them quite big, half a million variables and things like that.

And when you were at Monash you also worked on optimizing distribution of school results.

Of what? Of school results? Yeah, oh yes, I wrote the scaling program. I worked there with what I regarded as probably one of the best statisticians in Australia, Ron Bainbridge - long dead now - who was in the office next to me, he not only understood statistics in an academic sense, but he had a very real feel for statistics. He and I worked together to work out what it was we wanted to do. Everyone said we couldn't do it on the equipment we had then, but I figured a way of doing it. The main trouble with the scaling algorithm is basically the amount of data., it's a different program but the idea is still used, It is based on the fact that it was clear from the statistics, and all of the research, that teachers are very good at ranking the kids in their class, very good. They have absolutely no idea where those kids stand in relation to kids in another class. So, what you needed was to use the teachers ranking indications within the class, but you needed a way of scaling the class so that it would be at the same level as other classes. And we didn't have a lot of memory, it was only 64 K I think we were working with even then, but we had a disc drive, tape drives, lots of them, and I figured a way of shuffling the data between them and optimising that way. The trouble with it was that it was almost impossible to explain to arts professors who controlled the scaling. You couldn't explain it to them, it was too mathematical, and so there was a lot of opposition to that. It's still by far the fairest way of scaling the kids. There are no absolutes in that of course, but if you're in the middle of a very good group of kids and someone else is in the middle of a very poor group of kids, they shouldn't get the same mark as you. They tried to adjust for that. But this statistician was able to show, even before I got involved, that the pass marks from year to year were varying far too much for the number of kids involved. If you've got 30,000 kids, the pass mark can't change very much each year, because you've got the weight of all these and that averages out. You can show quite clearly that the scaling which had been done, well this is a pass mark, 43 is a pass mark, 42 is a fail. Now one mark in English, in the Victorian Matric as it was then, was about 500 kids passing or failing. It matters to get that scaling as fair as you possibly could. And then we did a similar thing with - again it was hard to explain - with allocating places in university, because you could submit your list of preferences but then certain departments would just ignore that. "If you haven't put us first preference you can't come", particularly in medicine. They just would not offer a place to someone who hadn't put them first preference. And I mean they've got more than one decent medicine school in Australia, so it was fair enough to put say, I don't know, Melbourne first, Sydney second, or whatever, or the other way around, but that excluded you from the others. And our system didn't allow that; the professors didn't like it a bit.

And you might have very legitimate reasons for putting one place second, doesn't mean that you're not a good student, so…

Well, yeah. One of the exams we did, The Commonwealth Scholarship Examination, which was Australia wide, and that was one of the early days when they used partly ranking and partly examinations, so partly normal examinations and partly rankings in school. And we had one kid, my staff at Monash Uni nearly went on strike (we had checks to make sure, because of course there could be keying errors when putting stuff in, it was all double keyed), and we found one kid who had been ranked by his school at the bottom of the class, and he was in the top 1 percent of the state in exams. So something was wrong, and they wanted a mechanism for us to query that, but the school had the final say. We got the message back: "This boy is a troublemaker; I will do nothing to help him to progress his academic career." If he wasn't a troublemaker before, he was going to be one after. Because when you're that bright you know you're bright, you know you're not the same as the guy who's failing everything, and he would see people in his class getting scholarships and he didn't. I don't

know what happened to him, I have no idea. But yeah, you're up against that sort of thing. In a sense, it's conservatism, and in a sense that's what you're up against in mining, a lot less now, a lot less now, but getting quite progressive in some areas but they'll still hang onto… One of the big differences in mining is that it's such a big industry that you have specialities, silos, where the guy who manages the mining, the physical digging up of the stuff, his salary, his KPI is based on mining as cheaply as possible per tonne. So how do you do that? You use as little explosives as possible, and you ship big lumps of stuff to the processing. The processing guy on a similar deal, has to grind everything down to talcum powder, he doesn't want big rocks, he wants little ones. He'd much prefer the miner to spend a bit more on the mining and he needs to spend a lot less on the processing. And those boundaries, you're up against it all the time, but again, the mining industry is starting to realise that you can't silo to that extent, there's got to be a common aim. And we've done a lot in the last 20 years in educating the mining industry in that sort of thing. Our son does what we call a strategic assessment, which is just $20,000 or something, it's two or three days work. But part of that is informing all the staff (and with Zoom it's a lot easier, you don't have to travel so much now), where you get the mining engineers and the processing engineers, and the sales people all in the same room - Zoom room - and they can all hear the arguments. And still some of them will say: "No, no, I've got to look after my bit", but that sort of thing is a barrier in mining. I'm sorry, I'm wittering on.

No, no, it speaks a little bit to my university background, what we're doing is looking at processes across organizations and what you're interested in is the organization as a whole.
That's right.

…not maximizing it for one of the, and I'm sure your algorithm has tried to optimize for the organization as a whole…
Yes.

So you were running into these social, political barriers that hold back the organization, because of the way it is structured.
Yeah, yeah, that's right. It's easier to manage if you break it into silos, so it's the managers at the top who decide, because it's much easier to manage if they can just talk about mining to this guy and just talk about processing to this guy.

That all makes sense. I think what has to change is the KPIs, right?
Yeah, yeah.

So that you encourage the behaviour as you learn that your behaviour has to change.
That's right, yeah.

Now I lost my train of thought a little bit.
You've got a train of thought? I lost mine long ago.

So what I wanted to say is it seems that the mining industry is very conservative and that you had a working product to show, was really helpful for you.
Yes.

Now looking back at what you achieved over the last 40 years would you do anything different with the benefit of hindsight?
Well from my point of view, not a lot, in that it was governed by the code: What can I do? How do I do it? but no, we wouldn't have done a lot different, well I'm sure we could improve little bits but no, I wouldn't, I think. Although it's taken a long time, I mean Whittle Consulting has been going 20 years and it's only just started to pay dividends, but we're not poor we've got shareholders obviously, and my wife and I have got about 40 percent of the company, so yeah. I can't think of anything major.

And in terms of support that you got from the Australian government?
Yes, that was important.

So that was important, so you would say like going forward Australia should make sure that they continue this kind of support through Australia?
Oh, yes, absolutely, I think it's very important, particularly when you're going overseas, well it has to do with overseas but when you're physically overseas the help that you get or got was invaluable.

Yeah, and in terms of development, did you get any support later for the development?
No. Well, that's not quite true, in the sense that as a company we can talk stuff up as research, development is research, and it's taxed differently.

So these kind of incentives for research…
Oh yeah, that's important.

To go into new products…
Yeah, yeah.

So we talked about the support and hindsight, did you build on any existing standards other than using FORTRAN and trying to make it as easy as possible for mining companies to use?
No, no, not that I'm aware of, no. There weren't very many standards in those days.

And looking back were there specific aspects that you think were outside of your control that had an important influence on your success?
Well apart from the conservatism, that obviously influenced, slowed everything down. We've got there now, but it's taken a long time. I don't think, well I can't think of anything, no.

And when you grew as a company did you grow first in Spain because that's where you had your first sale or…?
No, no, no.

…or was it just all over the place?
Well we grew in Melbourne first, obviously. We converted three of the kids' bedrooms into offices and had staff in our house. So we started in our house, outgrew that, hired an office just around the corner, outgrew that, had a bigger office, but it was all in Melbourne. And yeah, one of the things, mining engineers in the United States don't have the social standing that mining engineers do in Australia, and we offered, when we made sales overseas with

Whittle Programming we offered telephone support. But they couldn't phone us because they weren't allowed to make international calls. So, we set up a toll-free number and anywhere in North America you could dial 1800 WHITTLE and that rang in North Balwyn, at our expense of course. And then the kids found it and they'd ring us up and ask us what time it was, at our expense.

Yeah, that was the time of telephone pranks, that was a while ago.
Yes.

So you talked about hiring staff and that you tried to hire people that have different skillsets.
Yes.

So when you hire people what do you look for, is there something in particular that would…?
Well, it depends on what, I mean for tech services it's different from hiring, we haven't hired that many programmers, you don't need a large number, what you need are good ones, and so it's hard to say but we're always alert. One of the things, in the very early days, some of the part time people we got, I'd actually worked with elsewhere, I worked with at Newmont and that sort of thing, and they stuck with us for decades, because I already knew what they could do, what their strengths were, so I just chose them. We have very little turnover of staff, very little.

Any particular staff incentives or benefit schemes?
All our staff are shareholders.

And do you think it's important, what…?
No.

No, you don't.
Well, no, the most important thing is the atmosphere in the office, is the way people work together, and yeah, they're all friends. I mean we do have consultants overseas on contracts, quite a lot of them. But even they are chosen, at least in part, on personality and if possible, on previous experience of contracting them and working with them. But we certainly don't look for a jelly mould that we're fitting somebody into, we think more broadly than that. I mean running a big company with thousands of staff you can't do that sort of thing, but for a small company you can.

So when did you decide to offer staff to be shareholders in the business, was it something you did right from the start or…?
It was pretty early on, that was Whittle Consulting, not Whittle Programming, Whittle Consulting, because that's got shares. Basically, we don't normally pay bonuses, we give shares for bonuses. Salaries are a fairly steady increase, roughly in line with CPI, bonuses are on merit. So, once they've been with us 12 months all staff members are shareholders, in various quantities, obviously, but more shares are given out each year.

Yeah, and you say that's very important for the office atmosphere?
Well, it gives a sense of belonging. I mean in some cases the amount of money they get is not that big, but they can trade the shares between them, there are limits in the shareholder

agreement, there are limits how they can sell them. But yeah, I think they're pretty happy with those shares, because their face value is quite high when we give them, we're talking thousands of dollars for a bonus.

Yeah, so that is really an important incentive you're using for staff engagement.
I think so, yes. Again, I'm not sure this would be the same for a big company, I'm talking a small company, relatively small, sort of 50 staff in total. We wouldn't have 50 staff worldwide, would be 30 or 40 I think, at a guess. We're fairly small numbers but we work hard.

You only have a main office in Melbourne and the rest is consulting.
Yes, yes.

So you are involved in a number of professional bodies and societies?
What, I am?

Yeah.
Yeah, ACS and AusIMM, the computing and the mining, I'm a Fellow of both of them.

And what's the reason for you to become involved in those societies?
Well, ACS was funny because when it started way, way, way back in the 80s, or before even, I felt well, I'm programming I better be a member. I got the forms, and I just ticked a box and started paying my subs and what I'd ticked was Associate rather than Member, and many years later they wouldn't have a bar of me upgrading, because I was an Associate, and I hadn't done a course in computing. Hell, I'd given courses in computing, but there were no courses in computing when we started or very little, and there was a bit of a kerfuffle about that. Anyway, we got over that now, I'm a Fellow now. And you know about the AO, don't you, you know about the Order of Australia we got?

Yes, yeah…
Oh yeah, and again, although it's got my name on it. I didn't win it, we won it, always thought that way.

And in the mining industry you became a member because…
Well, because I was working in the mining industry, that's what we do. And actually that was interesting when I joined that, because I rang up the office, which was in Melbourne, and I said: I would like to be a Member and they thought a bit and said: wouldn't you rather be a Fellow? I said oh yeah. I was already pretty well-known in the mining industry. In fact I've often said that if you talk to a mining engineer anywhere in the world, if they haven't heard of Whittle they've been living under a stone, they really have. It's used, the stuff we sold is now used in hundreds and hundreds and hundreds of mining companies worldwide. I'd guess at least half the mines in the world have been designed using that software. I don't own it anymore, we get some royalties on some parts of it, but yeah, we're well known. At one stage my wife was better known than I was because she would do the marketing.

But 'Whittling' became a word before Googling became a word.
Well, 'Whittling', yeah, the mining engineers invented that, we didn't invent that, that's quite early days they'd talk about Whittling their pit. And indeed, what's the name of the banking company? One of the big international banks we talked to the guy from that once… he stated that they would not lend money to a mining company unless they were using the Whittle software. One of the huge international banks, that sounded nice.

Yeah. Now that you mention banks how was it initially to get enough finance to start the company, how did you do that?
We didn't, we did it out of our own pockets.

So you re-mortgaged your house and that's…?
No, it didn't really cost that much, because we were doing it all ourselves, and no, we didn't raise any loans of any sort to start Whittle Programming, and we still don't, we still don't borrow money.

So you could grow on the basis of your own turnover?
Yes, yes.

Now we're coming towards the end of the interview we always try to end up on a high note, are there any particular disappointments you in your professional career with the IT industry in Australia? I like to phrase it positively, opportunities to learn.
Opportunities to learn. I think the computing industry has done pretty damn well, actually, in Australia, after all it was pretty early in the game, we were building computers. I don't feel qualified to make these sorts of generalisations, observations, but I've had some pretty good contact with people.

Thank you. Do you have any further comments or observations that you would like to share?
Not really, no, no, I'm starting to slow down a bit, I am 92 and I am slowing down a bit, there's no doubt, and I don't program much anymore. But I've still got fairly good judgement I think about the way you program, and I try and give a little bit of guidance in that area, but no, I don't have anything more, no.

Then thank you so much for your time today Jeff.
You're welcome, you're welcome.

End