

Perforce to dream of SCM

Software Configuration Management (SCM) tools are getting a lot of press at the moment. Much of this is due to an increase in the amount of development going on and a realisation that process and management of source code is critical to success. The market tends to be dominated by the big players such as Rational and Borland (StarTeam). Despite this, Perforce believes that it should be given the same amount of respect as these large companies, so **Ian Murphy** set out to interview Christopher Siewald on where Perforce had come from, what made it different and where it is headed

It started really when I was working at Ingres. No matter what I did, I always seemed to get dragged into SCM. I was working back in 1985 in the porting group and trying to get the software out on different platforms. One of the people I worked with had a mantra that "Everything is an SCM problem". He was right, whenever things when wrong it was generally because changes had been lost or people didn't know what source code went where.

I wrote a simple application to manage code control and called it RRCS (Remote Revision Control System). A friend of mine wrote a little script on top called Piccolo and it worked really well. You could check-in and check-out and we had the notion of atomic change transactions. I learned that the best to handle stuff was when you submitted a bunch of files they should all go in with the same date/time stamp and the same change description. No point checking in stuff individually when you could check-in as a group. We used a linear log as the database which simply recorded everything that was done.

I went away and then return 5-6 years later and discovered that the system was still in use but working very slowly. This was because it was reading in megabytes of data every time it was used. So I decided, with some other people, to rewrite this and use a database underneath it. It became Piccolo 2 and we managed to get a fair amount of people at Ingres to actually use it. Prior to returning to Ingres I had been doing some consulting in Australia and considering starting my own SCM company. During this time I came up with the notion of Interfile Branching which is the branching model that we've been using in Perforce.

We spent about a year working on this and in 1993 we looked at how we could make this available to a wider audience. The idea was that if we let other people have access to it they might be able to do something with it. Around this time Computer Associates took Ingres over and everybody left.

A few years later a partner and I, one of the people who worked on Piccolo 2 with me, got together. We said "we've done this twice before so let's see if we can come up with another one". We've learned a bit more about SCM and Piccolo was now owned by Computer Associates. We've looked around and there really wasn't anything that did what we thought really solved the problem for SCM.

We wanted an implementation that was as clean as possible. I had learned a bit about databases at Ingres. I managed the networking group so I knew about networking and having worked for the porting group I knew about making things work on different platforms. We called the first version P3. Early on I found a copy of FreeBSD which delighted me because I've always preferred Berkeley Unix. We started coding around January '95 and one of the advantages of Perforce is that the core of the system all came out of one head.

Most of our competition is ahead of us in terms of size and complexity rather than coming in behind us. We're fairly comfortable that no large shop would ever be able to compete directly with us. That's because they would need a large team to design and build something like we've got. You can't get a small tight product when you have so many people working on it. A lot of key products out there are created by very

small teams or individuals but they get bloated over time as many people work on them.

We sold first copy of P3 to a customer in October '95. Although we called it P3 we knew we'd have to come up with a different name. We did the dictionary searches and wanted it be a real word starting with the letter P and the website also had to be available. In 1996 websites were going very fast. By August 1996 we became Perforce software and named all the components P4.

Since incorporation we have been profitable every quarter. We have to be. There is no external funding. Early on we did talk to venture capitalists but they wanted us to look at the web space and to get to be 10 times as big as we wanted to be. That's their model and we were just not comfortable with that.

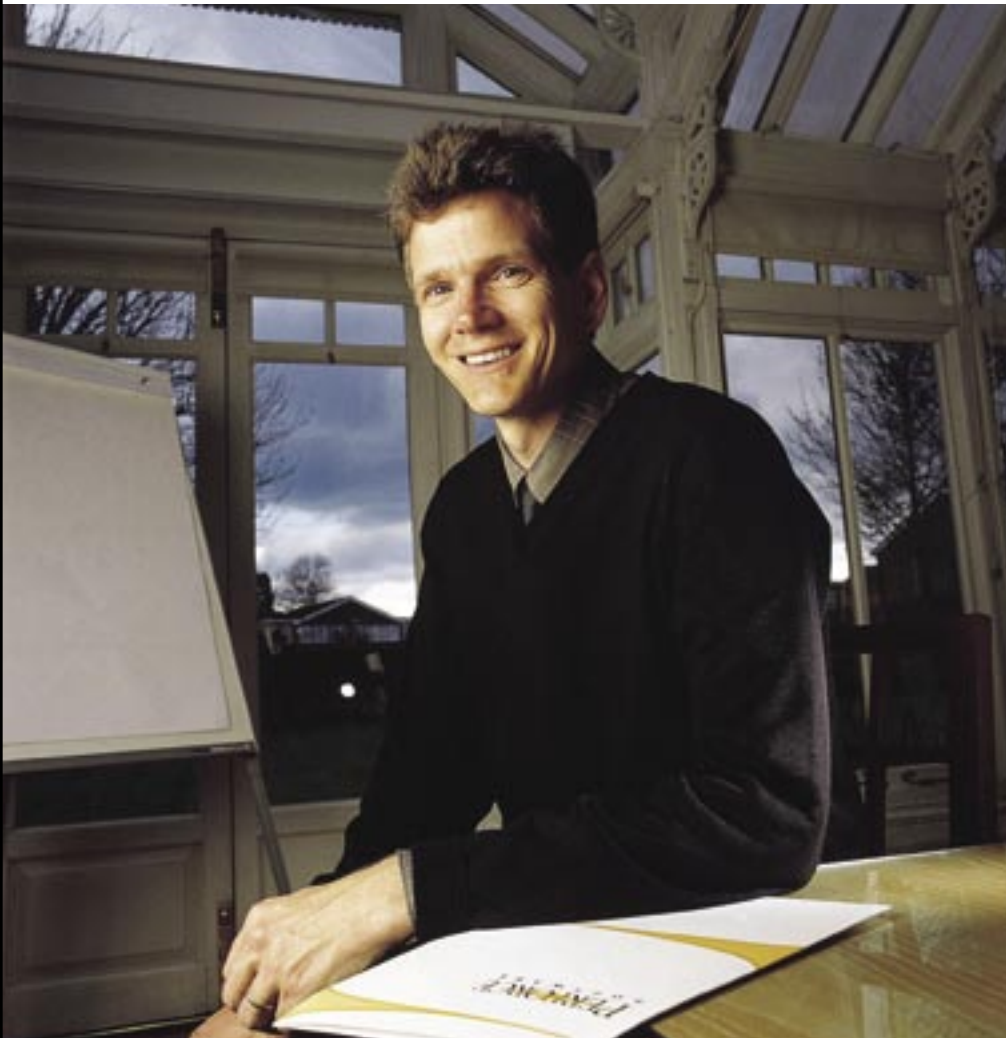
Decided to go low and slow, never overspend or live beyond our means. We could always spend what we've earned but nothing else. It's an approach that has served us well. We've hired at a consistent rate and now have 70 people and add 8-9 per year. Unlike other companies we've hired through both the slumps and the peaks. Our goal is to be staffed for the right size with the right people and not have to go out and chase people. As a result, even through the difficult times of the last few years, we never had to lay anyone off.

I was asked by a friend in the Bay area to do a Linux port in 1995 and we did. We got onto Linux very early on and are still on Linux. It represents about 20% of our servers out in the market. We still do our server development on free BSD. Portability has been our thing and early on we ran on Windows but were essentially a command line operation. One of our users said you need a GUI for windows and took some time out from his own job to code a prototype GUI front-end for Windows. He got it to the point where he showed it to us and we paid him to finish the job.

We call it P4 Win and still have customers using it today. There have been many changes of course but it works well. At the same time, someone came along and said we need to support the Mac. We wrote a port using MPW which was their shell environment. The result was that we were now on the Mac. Even today, we are still the only real player with clients that sit on UNIX, Linux, Windows and Mac.

Mac users wanted a graphical environment. We had a graphical interface for Windows but not Mac and UNIX and knew we needed to solve this. We started by taking a run at Java and it was a bad move. We ran into sharp propeller blades so we turned round and ran back. The supposed write once run anywhere environment really wasn't happening. This was at a time when Swing was being developed. It ran very slowly and we spent a lot of time tuning. The interface to C++ was poor and deployment on multiple platforms was as hard as writing it. We threw it all away - 6 months development.

We then decided to use the browser as the front end and wrote a small serverette which ran on the client. It was called P4 Web and was very good at browsing web content. There were/are a



lot of customers but we were still not hitting the mark. The problem now was that things were beginning to get very complex and I am not a fan of complexity.

Eventually we found a company called Trolltech who have a product called QT. It's a portable widget library and we used it to write the GUI front-end. It will adopt the native look and feel on a platform where possible and emulate if needed. About 1 year ago we released P4V for Mac, Windows and UNIX boxes. It's not as small and tidy as I would like but that's the nature of working with a widget library. Over the last year we have managed to get the performance and its functionality up to where we want it.

One of the things we realise is that performance is critical for developers. You are at the centre of their world and if you are slow, they will stop using you. We've been able to maintain our performance over time and scale the product. Early on we had customers with 30 users per server and now we have some with over 1000 users targeting the same server. We are the fastest thing out there and can scale to any size of project people want to manage in a single server.

So we are now the leader on performance and the only one usable on a large scale. We also started with portability as our primary goal and we've achieved that. Today we are still the only

real player across multiple platforms. The client/server approach allows us to manage local files using a central database. Files on the local disk means work at local speed but keep metadata centrally.

Always assumed some form of working network. Reliability of the net is important and that has served us very well. Customers have local and long haul networks, telecommuters and remote offices. Use perform over the internet. The strength of the model has served us very well. We have a model that reflects how people think of their source code. People get that aha moment when they see us.

There are a couple of mini-trends currently happening in the market. One we are riding on and one we are bucking.

Software developers have realised that they need SCM for years and this includes the fringe developers. But it's not just software developers that want SCM. Look around at the complexity of the market in general and you see that there are other groups that need SCM just as much. We do business with the embedded market but it's becoming so complicated many are simply becoming software developers today.

The same is true of anyone managing content either in a Web environment or in the office. Once you solve the SCM problem you suddenly realise that you lend yourself to solving other problems as well. You've done the hard part. As

my friend said all those years ago at Ingress, everything is an SCM problem. Legal documents are a good candidate for SCM because they get versioned all the time. Our EULA is managed in Performce.

We have just spun-off a company called IC Manage. They are building integration between Performce and Cadence for chip manufacturers. It allows them to manage both the chip designs and the software in Performce. nVidia is one of our customers doing just that.

We benefit highly from Microsoft and others talking about the need for SCM. They educate the market for us. When their customers reach the limits of their products we benefit because they come to us. We have a lot of customers who move from company to company and take Performce with them. We also concentrate on just selling to those people whose problems our products will solve. The goal is only to sell where there is a benefit and not chase the 5 user SourceSafe market. There's no real benefit to anyone there.

To determine customer loyalty we carefully track renewals. 75% of people who have bought Performce are still paying for maintenance in the last year. This means that they bought what wanted it and are still using it. These are the people we want to sell to. Of the remaining 25% of licences some are people who are happy with the product and don't renew support because they don't need it. Some licences are lost where people have downsized and some are now dead along with companies who no longer exist.

We've never had the case where a customer has told us they are ditching us and going elsewhere. Selling to people who want to buy from us is where we focus. A hard sell means hard support. If you push it down their throat and they will come back with lots of problems. We are lucky because people evaluate us thoroughly and then want to come back to use.

We make it easy for people to get started. There is a 10 minute pitch about SCM on the website and a ten minute demo to download and install in order to get going. We also have a lot of documents on the website. With SCM there is always a technical person doing the outreach for a new product and they are adept at looking at what they need.

The trend we are bucking is the move to integrated suites. There is a pendulum that swings between "Best of Breed" (BoB) and suites. BoB are doing well people start selling suites. The pendulum is still swinging in that direction and we are waiting for it to swing back. The analysts we talk to tell us that although people are selling suites customers are not really buying. Suites only work when you have all the answers and with workflow and other components in suites people don't have those answers. Another problem is that suites often need consultants and the more complex the suite the more expensive the consultants. BoB doesn't need a consultant.

What's exciting for us for the future is that people are now making that cultural shift towards SCM in all areas and we will reap the rewards for our work. ■■