



Presents
An IT Metrics and Productivity Journal Special Edition

Focus on Capers Jones, Chief Scientist Emeritus, SPR
A CAI State of the Practice Interview
July 2005

Biography of Capers Jones:

Capers Jones is Chief Scientist Emeritus of Software Productivity Research (SPR). Mr. Jones is the designer of several software cost and quality estimation tools including SPQR/20™, the first commercial software estimating tool to use function points as the basis for sizing source code and other deliverables such as specifications and user documents. He is also an international consultant on software management topics, a speaker, a seminar leader, and an author. As an author, Mr. Jones has written 12 books including his best seller *Applied Software Measurement: Assuring Productivity and Quality*. His most recent book is *Software Assessments, Benchmarks, and Best Practices* published by Addison Wesley Longman in 2000. For a more extensive bio of Mr. Jones please see www.unt.edu/isrc/Faculty/bios/JonesBio.pdf

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

CAI: It's been 20 years since we first heard about the SEI and the CMM from Carnegie Mellon. What is your opinion of the progress that has been made to date? What percentage of penetration do you see in large organizations to date?

CAPERS JONES: The CMM is not a panacea and it does not solve all problems. In fact, a case could be made that the CMM creates a few problems of its own. In general, however, the superimposition of the CMM structure on a good sized organization has benefited it wherever that has occurred.

Regarding why the penetration is still so low overall, there is no exact answer to this. However, if you look at other forms of human activity, such as military affairs, you will see that the adoption to improved technologies does not move instantly. It took 50 years, for example, before breach loading rifles replaced muzzle loading muskets. So just having something that is better than what was there before does not guarantee success. Social and sociological changes are necessary too.

Nevertheless, I think that the adoption of the CMM today is probably up among Fortune 500 companies somewhere in excess of 20%. And in the military and defense systems

world, the people for whom the CMM was originally intended, I would say that it may be over 50%. On the other hand, in the IT world, where people work on IT applications rather than technical software like weapons systems or systems software, the penetration is quite low. I suspect it's still probably 15% or less, but it's beginning to grow in the absence of any other equivalent method aimed at them.

CAI: You mentioned that in good sized organizations the penetration has done quite well. And we've heard a lot about the differences, from a CMM perspective, between military contractor sized organizations and ordinary IT organizations. Is there any way you could quantify for us what the actual size of an organization must be at which point the CMM becomes a viable strategy?

CAPERS JONES: Any organization with more than a total population of 1000 software engineers and professionals is a very good candidate for the CMM. And for the bigger companies, like the IBM's and the AT&T's, the companies with over a hundred thousand software professionals, these types of organizations should just adopt the CMM as a natural course. It's not a panacea and there are other things, such as Six Sigma and other methods, that they may want to explore at the same time. Nevertheless, only big companies can build big applications. And the CMM gives you a better control of the technologies of big applications than most other things out there.

CAI: Are you surprised by the number of organizations that are still struggling at Level 1 and Level 2?

CAPERS JONES: No, I'm not surprised at that, any more than it would be a surprise to see wooden sailing vessels 25 years after iron plated vessels became available. It's just that technology transfer and technology adoption is not always as quick as it should be. What made it quick, in the case of iron plated vessels, was the destruction of a lot of wooden vessels leading up to battle of the Monitor and the Merrimack. That got a lot of people's attention in a hurry. But we've had nothing like that. The CMM has not actually destroyed any other old fashioned methods yet.

CAI: For those IT organizations that have had a reasonably strong CMM program and CMM implementation, what would be your characterization— in aggregate— of the resulting improvements in productivity, professionalism, and effectiveness?

CAPERS JONES: Let's take a specific size of application so we can actually measure the benefits. Let's take 10,000 function points. And I pick that number for a particular reason. We do a lot of work as expert witnesses in law suits for projects that fail. And

of the 15 law suits that I've been involved with, all but one have been 10,000 function point projects or bigger. That seems to be the cut over point above which projects fail with embarrassing frequency and below which they can be put through to completion. So if you look at 10,000 function point projects, I would say that the solid CMM 3's and up are about 15% higher in productivity and 25% higher in quality. Additionally, these same organizations experience a 50% lower risk of failure.

CAI: Where is your data coming from?

CAPERS JONES: I haven't looked at the current numbers this week, but the overall volume of things we've looked at is something like 600 organizations and companies, 150 Fortune 500 companies, and 12,000 plus projects.

CAI: Given all of the advances in methodologies and CMM tools we still have data* pointing to the fact that IT software productivity has remained flat for 10 years. Do you have any thoughts about why that may be? (*source: IT Metrics Strategies@2001 Cutter Information Group)

CAPERS JONES: Sometimes companies go backwards as well as forwards. It's been kind of a drunkard's walk. What's happened is that the generation of people who developed some of these things– the Zachmans and the Watts Humphreys and so forth– they reach retirement age and go away. And the younger generation does not always pick up those solid lessons. So a lot of things that were being used by companies 15 years ago with some success have dropped out of use while new and more ephemeral things like agile methods have begun to surface. And these newer methods sometimes work and sometimes don't. But by the time the latter is discovered, the people who knew what did work have retired and gone away. Companies then have to relearn.

CAI: How does one raise awareness of these issues? Should the problem be addressed at the University level?

CAPERS JONES: I'm not sure that the University level is the way to do it. There seem to be more instructors in software related topics employed by Fortune 500 companies than at Universities.

A former Chairman of ITT once gave a public speech in which he said that, unlike electrical engineers and other kinds of engineers, it took about three years of remedial training after graduation before a software engineer could be entrusted with serious work. That was why ITT and AT&T and IBM and others had such extensive training

programs for their software communities and for people that were just out of college.

If you look at the way college courses are developed it takes close to 10 years before a new and proven technology can find its way into a textbook and into a teaching plan and then actually taught to undergraduates. First you need solid evidence that it works, then somebody has to write a textbook and put together the pedagogical materials, then it has to be put through various committees to get into the curriculum. And none of these processes move quickly.

CAI: Given that corporations spend 2-3% of their annual budgets on IT, what should a CEO's expectations be regarding what he gets for his money and how can CEO's measure this discretely? Additionally, how much of the software component of the average corporate IT budget gets wasted and how much of this waste could be quickly saved and recouped through the use of metrics, best practices, and IT management systems?

CAPERS JONES: I don't know every Fortune 500 CEO, but I've met a lot of them. And I would say that, as a class, the CEO's of Fortune 500 companies dislike and distrust their IT and software organizations more than any other part of their company. And the reason is that we have more failures and are harder to bring under executive and management control than any other group. As a result of this, they can't predict success. So we are in general distrusted and not fully regarded as being as capable of the rest of the company. Additionally, what most CEO's too often see are massive multi-million dollar failures of large IT projects and large system software projects and this consumes a lot of their time and energy. So generally speaking many CEO's are not happy with IT.

I can remember when ITT's Chairman Harold Geneen brought 40 people over from IBM, myself among them, to put up a big corporate research and education center for programming specifically because he was mad as hell at all the failures in software projects. He was willing to spend over 15 million dollars a year to fund a good sized group of close to 150 people to try to solve those problems throughout the corporation.

And other corporations have done the same thing. IBM for example made fairly heroic efforts to bring software under control by means of measurements. And I would say that, at least in the late seventies and eighties, they were doing a pretty good job.

It's in the companies where the CEO's have either given up on software or where the whole management community has given up that you will see the biggest problems.

CAI: Would you like to take a stab at quantifying how much of the average corporate IT budget gets wasted?

CAPERS JONES: I actually have fairly explicit data on this. I wrote a paper on wastage

(The Impact of Poor Quality and Canceled Projects on the Software Labor Shortage, July 12, 2005). The gist of that paper is this: close to 50% of the total man hours spent on software in big corporations is wasted. Keep in mind that the word 'waste' has a special definition here. It either means fixing bugs that should not have existed in the first place or working on projects that are so badly run that they are cancelled and never reach their intended customers.

Nevertheless, small projects below 1000 function points generally zip right through to completion. It's the failures on massive projects, the 10,000 – 100,000 function point projects, that are absorbing most of the wasted hours.

CAI: When you talk about the smaller projects, has anyone really measured the amount of rework that occurs here? Our indications are that there is still an enormous amount of rework involved in these projects.

CAPERS JONES: Small projects do have defects. However, a project of 1000 function points in size will probably total about 4 bugs per function point during the course of development and around 90% of them will be removed before the customer sees them. If you go up an order of magnitude to a 10,000 function point project, there you are looking at around 6 bugs per function point, and you're not going to get rid of more than around 80% before the customer sees it. So you have a reduction in removal efficiency. And that drop from 90% to 80% has a disastrous effect in terms of what the customer sees. That's why you will get a lot more grief at the big system level than the small. And if you go up to 100,000 function points, where you start approaching 8 bugs per function point, now you are down into the seventieth percentile.

Now, there are technologies that can change this. There are things like the old, traditional, structured methods, there is joint application design, there are a host of things that can reduce defect potentials. And there are other things like formal inspections, which were invented by Fagan way back in the sixties, that can raise the removal efficiency to over 95%. But part of the problem here is that not a lot of companies know about these things. Or if they know about them they don't deploy them. And that gets back to the social issue of why these things, which were proven to be successful thirty years ago, are not as widely deployed as they should be.

So the main problems facing the software world can actually be expressed in a relatively concise manner. The first issue is bugs and the need for careful quality control. The second issue is the rate at which requirements grow during the development of a project. That rate, the rate of requirements growth, is about 2% per calendar month. Nevertheless, a lot of big projects don't deploy successful technologies for controlling the rate of requirements growth during development.

These two issues alone—the number of bugs and the rate at which requirements grow— probably account for 80% of all the project failures that we've ever seen.

CAI: Our sense is that the number of silver bullet methodologies and technology solutions that have occurred over the past 20 years have tended to make these problems more complex. It seems that we keep trying to learn new things faster than we can make them work well.

CAPERS JONES: I agree. Instead of going with a solid technology like inspections, with 35 years of empirical data and a proven track record, people tend to jump at the newest things, as in the case of agile development methods.

Agile development methods are actually quite dangerous over 1000 function points because they are too sloppy during the requirements and design phases to successfully hold up the infrastructure of a big system. But because it is a new thing and a lot of people don't realize that it will fail, and because they try it out on some small projects and it seems to work, they will jump into a big system with it only to discover, to their horror, that they're not getting the kinds of results they were getting at the low end. This is basically just a combination of inadequate training in proven technologies along with the basic human instinct to jump at the latest thing. I would agree that this is a problem.

CAI: If the manufacturing revolution allowed the average manufacturing company to increase productivity by 400%, what is the opportunity in IT?

CAPERS JONES: There are two different channels of proven productivity. The first involves outsourcing. By using outsourcing organizations in other countries as an extension of the American workforce the work on a project can be transmitted to some other country during the night hours. In this case, what you are doing is stepping up to 24 hour development. That shortens elapse time. And the companies that are doing this are shortening their elapse time by 60-70%. But you have to be pretty sophisticated to do this.

Now the other thing that has the greatest potential for productivity gains is full scale reusability. But I don't just mean reuse of source code. You also have to reuse test libraries, specifications, user documents— you have to reuse a lot of things. But reuse will only have a positive value if the defect levels in the reusable material are close to zero. If you're reusing something that's filled with bugs, you will lift your maintenance costs to scary and uneconomical levels and the whole thing will become a failure. So in order for reuse to work you have to be very sophisticated with quality control and you have to have a certified library of reusable components and only a handful of organizations have stepped up to this.

This is a very hard thing to do. It takes really smart people with really good management support and really good technologies and there are not a lot of places that have these things in combination.

CAI: So much has been made of the cost differentials between India and the US, but Deloitte reports that there is only a 10-20% real cost reduction possible through offshoring. What do you attribute this to?

CAPERS JONES: The bottom line with offshore outsourcing is that whenever a country becomes successful at doing something, the salary levels go up for their people until they reach the comparable salary levels over here. So outsourcing to a low salary country cannot be effective for more than a few years because their success is going to drive up their salary levels. In light of this, I would say that the most you can get out of international outsourcing is a ten year window of success. After that point the salary differentials will begin to disappear.

In fact, if you are looking for long term benefits in terms of reduced salaries, you'll be running out of countries pretty soon. Japan is so expensive that you can't do anything there anymore. And India is already starting to see significant inflation in their salaries. And most of the former low income countries of 15 years ago are now fairly prosperous and moving up the scale. And some of the countries that still have low incomes, like Bangladesh, don't have the technology base or the infrastructure to do the work. So I think that outsourcing is at best a transient improvement.

CAI: In your opinion, what are the critical success factors for a successful offshore operation?

CAPERS JONES: Finding a CMM Level 3 partner or higher is going to be key. You will also need a company that is financially solid and that is not going to go bankrupt on you. And these days you will need a very good security system to deal with the possible disruption from terrorist attacks. And you also need to be very solid with the materials that are sent over—the requirements and the specifications and also with the work plans. Finally, you have to be sure that the quality control steps used overseas match the state of the art levels used in this country. In other words, you have to be pretty sophisticated on your end to make the offshore outsourcing successful on their end.

I would like to additionally note that part of the diligence done in selecting an offshore outsourcing partner involves going beyond an anecdotal understanding of their CMM level. That's because the CMM level can be used to rate divisions within organizations and even just projects. It is not necessarily meant to apply to an entire organization. And of course, if you don't understand clearly what their CMM rating pertains to, you can run into problems. You can suddenly find yourself in an organization that has many different CMM levels within itself and you may not be dealing with the highest one.

CAI: That leads nicely into our next question. We've talked to a lot of CIO's about offshore outsourcing and one of the things that we have heard over and

over again is that, had they had the rigor in place that was ultimately required to go offshore, if they had had this rigor before deciding to go offshore in the first place, they probably wouldn't have needed to go offshore.

CAPERS JONES: I think that is a very solid observation and very well put.

CAI: Final question: if you had to take a 500 person domestic IT operation at Level 1 and recommend a generic plan of action for improvement, what would be the first 4 or 5 things that you would have them do immediately?

CAPERS JONES: One of the first things any organization should do that wants to improve is to measure exactly where they are when they start, in terms of both productivity and quality, in order to give themselves a baseline against which improvements can be measured. And this is especially good if you are bringing in consultants or outsourcers because you may have their contracts written under the assumption that there will be gains in quality and productivity. And if you don't know where you started it will make it difficult and possibly litigious to handle their contracts. So that's why you need to start early with a measurement program.

Another thing you will need to do is bring quality under control. And that means formal design and code inspections, formal testing, test library controls, and probably even the creation of an independent quality assurance group to gather the data so that the data can't be smudged or distorted by project managers. You should also be looking at the rate at which requirements change because changing requirements mid-flight is the second biggest source of grief on software projects.

If you can handle these two things—measurement and quality— then you are addressing two of the biggest causes of disaster.

[Mr. Jones can be reached at: *Software Productivity Research LLC, 1734 Brevard Road #183, Hendersonville, NC 28791*. His phone number is *781-273-0140* and his fax is *781-273-5176*. Mr. Jones can be reached by email at: *CJones@spr.com* or on the world wide web at *http://www.spr.com*]

Questions? Suggestions? Comments? Please contact the IT Metrics and Productivity Journal Editor at michael_milutis@compaid.com.