



Presents
An IT Metrics and Productivity Journal Special Edition

Focus on Dr. Robert Charette, Master Risk Management Practitioner
A CAI State of the Practice Interview
March, 2006

Biography of Dr. Robert Charette

Dr. Robert Charette is an internationally acknowledged authority and pioneer in information systems and technology, systems engineering, risk management, and the lean development & management of large-scale software-intensive systems. He is currently President of the ITABHI Corporation, an international high technology company involved in information and telecommunications systems management consulting.

Dr. Charette is also on the advisory board of the Project Management Institute's Special Interest Group on Risk Management. He has served, additionally, as an elected chairman of the US Software Engineering Institute Risk Advisory Board (1995-1997), as a member of the National Research Council's Review Committee of Space Shuttle Software Safety (1992-93), and as Vice-chairman and Chairman of the National Security Industrial Association Software Committee (1988-89, 1990-91). He is currently on the editorial board of Software Quality Professional magazine.

Dr. Charette is the author of *Software Engineering Environments: Concepts and Technology* (1986), *Software Engineering Risk Analysis & Management* (1989), *Applications Strategies for Risk Analysis* (1990), and *Introduction to the Management of Risk* (1994). He is also the co-author of *A Unified Methodology for Systems Development* (1987). Our interview with Dr. Charette took place in November, 2005.

◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆ ◆

CAI: Could you tell us a little about yourself, how you got started, the path your career took, and what you are working on today?

ROBERT CHARETTE: After I left the US Air Force in 1974, where I was an avionics and

electronic warfare technician, I returned to the University of Massachusetts and completed my undergraduate degree in electrical and computer systems engineering. I went to work in January 1976 as an electronic engineer with the Naval Underwater Systems Center, now called the Naval Underwater Warfare Center, in Newport, RI.

As an electronics engineer with NUWC, I worked on a whole host of real-time software systems. These included the various combat control systems, sonar systems, and intelligent systems employed on different types of attack class submarines. I also had the opportunity to serve in a several different job positions, ranging from hardware and software engineering and project management.

One of the interesting things at that particular point in time was that the government was having a hard time getting computer people to go to work for them. Consequently, if you had any computer background at all, the government basically gave you the opportunity to pursue whatever you wanted. So I got to work in lots of different areas with a lot of responsibility for somebody who was then a pretty junior person. In effect, I received twenty-years of experience in less than ten years of time.

One of the jobs I had toward the end of my stay with NUWC involved the evaluation of combat systems, which included looking at future combat and sonar systems for attack submarines. This was one of my first experiences with applying formal risk management and scenario planning. Naturally, these combat and sonar systems were all software driven, but implementing them was not entirely a software issue. What we were really trying to understand at the time was how you could better link strategy to technology.

CAI: Could you elaborate on that?

ROBERT CHARETTE: As we were building all these systems, we were, to say the least, very technology-driven. We were driven by a technology imperative: whatever the latest and greatest “new thing” was, we pursued it.

Nevertheless, there was always this nagging feeling in the back of my head that we were pursuing clever technological ideas – and they were pretty neat from an

engineering perspective – that really didn't have a connection to Navy strategy. In other words, creating clever technology was becoming more important than being able to support the Navy's submarine strategy.

I subsequently spent quite a bit of time trying to understand what we we're trying to accomplish in terms of Cold War attack class submarine deployment and how this strategy meshed within the overall national strategy. I took courses at the Naval War College, read a ton of books, and talked to people in the fleet in an effort to understand what it was all about. I then forced myself to draw a direct link from the technological capabilities we were creating in our combat systems to specific strategic and tactical goals, in order to find out where the gaps were.

Consequently, one of the tenets that has really driven my risk management career over time has been this obsession with understanding the link of technology back to strategy. How should the two inter-react? For example, should strategy push technology or should the technology push strategy? How do you balance things out if one or the other starts to push too much in the wrong direction?

This is a very important subject because in my opinion there is still not much thinking that goes on in the IT community regarding the link between strategy and technology, and all the elements that can help or hinder that link. We still have all these organizations jumping on the latest technologies without first bothering to figure out what is it that they really want to do. We have to keep in mind that when we start a project, when we specify something like a software requirement, or when we pick a technology to implement it with, we are essentially creating a potential future. We are creating a vision of ourselves through the systems we build. Yet once this project is underway or we implement a particular software requirement or build on some technology, we may or may not have created the future we really wanted. The result may wind up shaping a future which was totally different from what we originally desired. As my old boss at NUWC used to say, we may get what we want but not what we need.

CAI: How did you eventually enter the private sector?

ROBERT CHARETTE: After I left NUWC, I went to work in 1984 for a small company called SofTech which was headquartered in Boston. SofTech was known at the time for its very advanced software and systems engineering work. These were the people who helped define the ADA programming language and its support environment. I worked at SofTech primarily in a systems engineering function, evaluating how organizations could better manage their software and systems engineering development processes, methods and tools. Creating systems engineering and software engineering methodologies was a big deal back in the late 1970s and early 1980s. I also got involved with some advanced weapons and logistics system design for the Air Force and on the early software support system designs for NASA's space station program.

In early 1986 I got recruited by the UK government to do risk assessments for a very large scale IT system development that they had underway: the computerization of their Department of Health and Human Services. This was around the time I finished my first book, which was focused on the problem of how to design, build and support software on a very large scale. The recruitment led me to London, and in turn, to Computer Sciences, Ltd. where I worked under contract for a year. This enabled me to get my required UK work permit. The contract also helped me start my current company, ITABHI.

I ended up actively consulting in the UK and across Europe for almost a decade, working with various governmental departments such as the department of Health and Human Services, the Home Office, the Ministry of Defense, as well as commercial companies in the telecom, computing and utility industries. I was also consulting back in the US as well, which kept me on airplanes a lot of the time going back forth across the pond.

During this time I also wrote a couple books on risk management, one of which focused on software engineering risk analysis and management and the other on application strategies for risk analysis. These books helped get me hired in the early 1990s -under contract to the UK Central Computer and Telecommunications Agency (called the CCTA) - to write the UK's first risk management guidelines.

I also started getting involved at this time with a lot of different commercial and government organizations on a lot of eclectic assignments covering a wide variety of

risk management topics. For instance, I served on the post-Challenger assessment team for the National Research Council, looking at software safety and software development process used for the shuttle. I chaired the National Security and Industrial Association (now called NDIA) software committee. I chaired the SEI's risk advisory board. I sat on a board of advisors for a group assessing the state of IT security at Big 12 universities. I was a founding member of the PMI risk specific interest group. And just recently, I've had the opportunity to serve as the ISO and IEEE standard's chair for software and systems engineering risk management.

CAI: How would you define software risk management? Can you break it down into key components for us?

ROBERT CHARETTE: I look at software risk management as I look at risk management as a whole. It's all about making high quality decisions through high quality risks. Risk management is just an element of decision making and software risk management is just a sub-component of systems engineering risk management which, in turn, is a sub-component of business or enterprise risk management. I don't believe that you can say there's something called "software risk management" without also saying that you're involved in systems risk management or in business risk management. The three are really intimately interconnected. This gets back to the linkage between strategy and technology.

To understand this linkage, you first of all need to worry about principles. What is it, for instance, that you are trying to accomplish in terms of managing risk? What are the things that you really value as an organization when it comes to risk? For instance, are you a risk-taking company or a risk-averse company? Which one will determine your risk tolerance? What are the things that you value as an organization in terms of managing risk? For example, is open and honest communication of risk a core organizational principle?

The second thing you need to worry about is the risk management process itself. What do you need to have in place that is visible, repeatable and measurable in terms of a risk management process?

The third thing is behavior. Behavior is critical because when we make choices, we intend to act. When it comes to risky situations, we need to think about what we want people in our organization to do as well as what we don't want them to do.

When you're looking at doing risk management, you need to create a principle-based, process-focused, behavior-driven system - a system in which any two of its pillars support the third. For example, the principles and processes you create must condition the risk taking behaviors you desire within your organization. Similarly, the behaviors and processes should reinforce the risk principles you value.

You have to understand what it is that you want people to do when they're faced with risk, when they're faced with something that may make it look like they're not going to succeed. You have to convince people to look at things differently than they normally do. But within a framework; otherwise you risk making the kinds of mistakes that ensue from more ad hoc approaches.

What's always been intriguing to me about risk management is that, superficially, it appears extremely easy. You simply look at what might occur, you clarify how these things might hurt you, and then you develop some approaches to keep the bad outcomes from coming into being.

However, while the process itself can appear to be quite superficial, it quickly becomes extremely subtle and complex. That's because risks are perceived and "unreal." They are only possibilities. They are not actual things. By the time risks become actual things, they are already problems and at that point they cease to be risks.

Furthermore, when you try to manage or reduce these probabilities, you quickly come up against a perplexing problem: if one allows resources to be spent on the reduction of risk, will the probability of project success be increased or, in fact, reduced? In other words, if you are spending finite resources to reduce mere probabilities, things that might not happen, couldn't (and shouldn't) you also be applying those same resources to things that you know really are happening?

As you can see, the process can quickly become very messy. It is often quite counter-intuitive, and that is something that is eternally fascinating to me, because we're dealing here with potentialities, with trade-offs, with futures and to really understand

what's going on with these intricacies, your thought processes have to be broad, deep, and quick. It's like the old saying "cheaper, faster, better - pick any two." In managing risk, you often can identify and prioritize the risks, but you may not be able to mobilize to actually deal with them. And if you manage to mobilize the right resources to deal with one set of risks, you will simultaneously be making a conscious choice not to mobilize against some other set of risks, which effectively means that you are accepting those risks. In this respect, knowing what your opportunity costs are is going to be key.

CAI: The Standish Group reported in 2000 that over 70% of software projects undertaken by large small and mid-size organizations came in over time, over budget, or not at all. What is the relationship, in your opinion, between the practice of risk management and these success and failure rates? Do you think that risk management can and should be used to address these types of problems?

ROBERT CHARETTE: Risk management is very important for creating successful projects. Effective risk analysis and management will help you identify what your assumptions are, what your constraints are, what your real objectives are and what can go wrong. Effective risk management will also highlight the perspectives and expectations of the various project stakeholders involved. It is a superb tool for bringing issues to the surface that traditionally get glossed over.

That being said, I think we need to make a distinction between project failures and project blunders. Many of these so-called project failures are actually in my experience blunders. A blunder is when we don't do the things that we know we should be doing and that we are also able to do. We know how to create successful IT systems, for instance, but in most cases we simply don't bother to apply the practices that are out there for increasing our project success rates.

Moreover, if you look at the relationship between risk management and project success - and Dr. Bill Ibbs over at UCal Berkeley has been doing a lot of research in this area - you will find that risk management is the least used of all the project management

disciplines. Not surprisingly, it's least used in the IT community. The IT community simply does not apply risk management effectively. The corollary, of course - and this is reflected in Bill's research, too - is that those organizations that do use risk management tend to have a higher level of project success than those that do not.

CAI: Would you be able to quantify the percentage of IT organizations that are using risk management practices properly and getting positive benefits from them?

ROBERT CHARETTE: One of my side jobs is that I am the Director of Enterprise Risk Management and Governance for the Cutter Consortium. We did a study back in 2002 that took a look at organizational risk management practice. To answer your question, we found that 51% of the organizations we surveyed claimed to be using some sort of formal approach to assess or manage risk. In other words, 51% had some sort of repeatable process that they were following. Nevertheless, of the organizations we surveyed, only 39% were applying software risk management practices. In fact, risk management was still a fairly new practice within the organizations we surveyed. On average, we found that companies had been using risk management for only 4 or 5 years. Consequently, program and project risk management has yet to be integrated into a corporate approach to managing risk.

What was interesting, though, is that although there was a minority of people using software-specific risk management practices, 90% of the people in our survey agreed that managing IT risk was either important or very important for achieving project success. In fact, 75% believed that software risk management made their projects more successful than projects that didn't employ risk management practices.

However, if I were to refer simply to my own personal experiences, I would say that the number of people who are using software risk management practices effectively is probably in the 20-30% range, and I'm probably being optimistic here. One of the problems that I regularly encounter - even in organizations like the Department of Defense, where risk management practices have been mandated now for almost thirty years - is that although a large number of organizations are "using" risk management,

its practice is really just pro-forma. In other words, they're applying a "tick-in-the-box" risk management process, and it's not affecting organizational decision making in any way.

CAI: Could you highlight for us what, in your opinion, might represent the top three software development risk factors?

ROBERT CHARETTE: How about the top 100?

The primary factor I see is the lack of realism. Our industry likes to over-promise and under-budget. We seem addicted to unrealistic objectives and unrealistic goals, even in the face of very complex projects. We pretend that we know more than we do, and then feign surprise when things don't go as planned.

The second major factor lies in the fact that, as an industry, we tend to be very sloppy in terms of our development practices. If you take a look at the Software Engineering Institute's CMM or CMMI results, you will see that the vast majority of organizations are still employing undisciplined or chaotic development practices. Poor project management practice is pervasive throughout the development world today. And poor project management will take a project down faster than any other type of risk factor except the lack of realism.

The third major factor revolves around politics. Projects do not sit in an objective, purely rational vacuum. They are part of a greater whole, one involving the political realities of an organization. Most people don't manage organizational politics well, nor do they recognize their importance to project success.

Each of these three factors, and there are certainly more, must be examined, understood, and managed in a very aggressive, realistic, holistic and honest way. And we should not ever underestimate the importance of honesty. We must always ensure that our objectives are both realistic and honest. The paradox with honesty, of course, is that you might have a hard time getting your project supported if you are totally honest about the risks that exist. The temptation to over-promise is rooted in this paradox. To be unrealistic, however, is to court disaster. That is far worse.

What this means is that until we get a development environment both at the business end and at the technical end, a fact-based environment in which we can be honest, then all of our risk factors will just be exacerbated.

CAI: Once identified, organizations could spend years investigating their own risk items. In light of this, what is the most practical approach for proceeding, once the risk identification phase has been completed?

ROBERT CHARETTE: There are two things you need to do. First, you need to prioritize your risks. Second, you need to mobilize against them.

Regarding prioritization, there are two simple questions that you can ask yourself here: 1) what is going to hurt me the most; and 2) what is going to hurt me soonest? You must deal with these risks right away; specifically, the ones that are going to keep you from accomplishing the next milestone or the next objective in your schedule.

Regarding mobilization, and this is an area that people tend to forget about, you must remember that a risk hasn't gone away just because you've allocated resources to try to avert it. You're not done until your mitigation strategy has actually accomplished its goals. So once again, in the short term, attack those things that are going to cause the most amount of damage to you soonest. Second, be aware of the great danger posed by the medium risks, too. The medium level risks are the ones that really can hurt you because they're the ones that you tend to accept. And if you have enough of them, they can overwhelm you. The worst thing in the world, in my opinion, is having lots and lots of medium level risks on your project. I'd much rather manage a project that has lots of reds and greens; don't give me one with lots of yellows.

Finally, keep in mind, despite their very low corresponding probability, that there are still some extremely high consequence risks that may be able to take you out. Keep a close eye on them.

CAI: How would you characterize the importance of tools in all of this? Do you think an organization can be successful without making significant

investments in this area?

ROBERT CHARETTE: Yes, I think so.

The majority of organizations already own a significant number of tools to help them manage risk. It's just that they don't look at them as risk management tools. They might look at them as information tools or planning tools but any one of them, in conjunction with the application of a properly crafted risk process, could be turned into an effective risk management tool.

Where specifically crafted risk management tools start to become more important is when you have larger projects where there are many different stakeholders. A good example of this would be with defense or large telecommunication-type projects. In a defense-related project, for example, you might have 20 or 30 different key suppliers. At that point, it's pretty difficult to manage the level of project complexity, to manage the large number and kinds of risks, without having at least some type of standardized risk capturing and communication tool. But even so, the tools are only going to help you with the paperwork component involved in the assessment and communication of the risks. In my experience, risk management is still a very intensive thinking process that is not easily automated. I would say that 95% of risk analysis can not be done with tools.

CAI: How would you characterize the use of estimation models for predicting the impact of risk factors on a project budget and schedule? How are these things inter-related, if at all?

ROBERT CHARETTE: Right now, they don't work very well. Estimation models haven't really been designed to help you look at project risks.

If I use an estimation tool, and it tells me that no one has really been able to do this type of project before, then I think I can safely assume I am not going to get there. That's what these estimation models do well.

What they don't do well is the linkage identification between all of the various risks. As a result, I still won't know whether or not the resources that I have allocated are

sufficient, or whether or not they're allocated in the right way. Nevertheless, if I can leverage my risk assessment to help me model my estimate - by looking at the level of uncertainty involved - the estimation tool starts to become much more valuable to me. But estimation tools by themselves are not very useful. I really need to have the risk assessment first, to help me parameterize the uncertainty in my plan, in my schedule and in my objectives.

CAI: How would you characterize the importance of processes in all of this? From a process perspective, what in your opinion are the critical success factors for effective software risk management?

ROBERT CHARETTE: First of all, you need to have some measures. You need to have information that is fact-based or evidenced-based. Whether or not you call them software measures, or performance measures, it doesn't really matter to me. What I'm interested in is having something that I can objectively measure against, and then predict against. I also need a process that's going to help me evaluate not only my objectives, but also my assumptions and constraints. We tend not to look at our assumptions. One of the things that I frequently tell organizations is that if they can't perform a full-blown risk assessment they should at least conduct an assumptions analysis, because it's the assumptions that underpin your project. You need to constantly test those assumptions against reality.

For instance, one of the assumptions used by most organizations when they estimate is that all their programmers are above average. That's just one example. We also tend to make assumptions that our budgets are going to be fixed or that we are going to get the funding profile that we planned for. There are a lot of issues like this, and we need to look at them very carefully, because one of the central axioms in risk management is that any assumption we make is a risk we accept. In this respect, our assumptions may be the most important things for us to be examining. Consequently, if we are going to do just one thing from a process perspective, we really should be doing an assumption analysis.

CAI: It's funny that these models assume all programmers to be above

average. That's kind of odd. It seems illogical.

ROBERT CHARETTE: I have never seen any estimate of any project - not one - that has said "our programmers are below average." I have never a single project estimate in my career that has claimed "our programmers are only average."

This may seem illogical but it gets back to the subject of organizational behavior. Realistically speaking, if you are in either an external or an internal bidding situation, are you ever going to admit that your organization is... just average?

This is exactly what I meant when I spoke of the importance in risk management of being realistic and the need to understand human behavior and politics. Rarely, in fact, will you find somebody without any skin in the game - an impartial observer - who will come to the conclusion that everybody in an organization is above average. There are very few organizations that consist entirely of above average people.

But what we see, especially in the IT community, is this peculiar idea that everybody is above average. I like to call it the Lake Woebegone effect. And we do it not only with our people, we do it with our schedules, we do it with quality, and we do it with requirements.

This phenomenon of the "above average programmer" is ubiquitous and costly. If you say that your programmers are above average when they are in fact average or worse, you end up putting yourself into a situation where, as your project begins to start, you are already behind. But you're not really behind, you're only perceived as being behind because you established unrealistic expectations that couldn't be met. The moment you find yourself facing expectations that can't be met, how do you get back on track? By taking increasingly risky decisions. You take shortcuts. You try anything that will buy back time. Well, if you start gambling with risky decisions like this, what's the probability that you are going to come out on top in every one of those decisions?

This is the reason I believe that the real value of risk assessment and risk management today, in the field of software, is the extent to which we can bring realism back into the picture. In my opinion, this is where we will find its greatest value. There is nothing overly esoteric about this. It may just mean somebody speaking out and saying, "How is it possible for everybody to be above average?"

CAI: You mentioned the importance of having measures, of being able to objectively measure against things in order to develop a starting point. What do you think of the relative value of external versus internal benchmarking data?

ROBERT CHARETTE: You have to get information from the inside of your organization. Set up your measurement programs, start getting data, and at that point compare what you have with the external world. I should also mention that I rarely see organizations that have effective risk management programs in place without also having very effective measurement programs as well. It's kind of a chicken and egg problem, though. Do you start with a measurement program first and a risk management program second, or vice versa? I'm not sure, but if your leadership is willing to simply state "Where are we?" that's a good start.

CAI: For anybody who might be new to the subject of software risk management or who might simply be interested in further reading along these lines, could you recommend a few books?

ROBERT CHARETTE: For software risk management, a good place to start would probably be Ed Conrow's "Effective Risk Management." You could also look at Tom DeMarco and Tim Lister's "Waltzing with Bears." Capers Jones has a book on software risk management, too. I would recommend my own books; for while my examples may be a bit out-dated, the principles are still applicable. You could also take a look at the ISO and IEEE standard on software and systems engineering risk management.

However, these are only starting points. If you want to be good at software risk management, you really need to pursue a broad and deep understanding of risk and its management from multiple viewpoints. You really must become a "student of the game."

In risk management, what you're really talking about, at the end of the day, is the

simple act of making conscious choices, which is something that sets us off from the rest of the species occupying this planet. It's a distinctly human endeavor. Consequently, you have to understand not just the mechanistic processes involved in risk management; you must also take into consideration the behavioral side, the human side that is involved.

I've been working in this area for a long time, almost my entire career, some 30 years of making trade-offs and choices as an engineer, and I can honestly confide that I don't believe I will ever get a complete handle on all of the subtleties that are involved in this game. However, I can also say that I learn something new just about every week in this field, which always keeps it fresh and interesting. Something is always happening somewhere which sheds some new insight into risk and how it can or should be managed. Whether it's what happened in Katrina or what is happening now with the flu pandemic or with the Iraq war, all of these seemingly unrelated world events, if you pay close enough attention, can ultimately help you understand how better to manage risk.

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