

Behaviorally Speaking: – *SCM Patterns – are they really practical?*

Bob Aiello – Apr 2004

This month I made myself extremely unpopular with my fellow editors at CM Crossroads by suggesting that SCM Patterns were overly complex for practical application in many large scale business environments. It shouldn't come as a surprise that my comments were perceived by a couple of our more accomplished Editors as being fairly close to CM heresy. I am not good at writing about something unless I really believe in it. I carefully read through the comments offered by my colleagues and read (and reread) their book that deals extensively with this topic. Part of my negative feelings may be a result of wishing that I had written down (and published) these paradigms myself. But another part of my hesitation is the realization that SCM can often be inherently complex and require complex tools. SCM Patterns is certainly a powerful methodology and well worth the effort to explore its application in everyday SCM practice.

Did you say that there is a Pattern Language?

I admit that I am not very visual and I don't really like anything that requires that I study pictures. I have written previously that I grew without much visual acuity (although today I can see fine thanks to wonders of modern ophthalmology). I got through High School and much of College with only being able to see shadows and large print (and then only for a short time). I really find the SCM Pattern Language confusing. But still I stayed with this and reviewed the material a few times. It might be tempting to just suggest that is my own personal problem, but a quick survey of a few colleagues of mine confirmed that there are other CM Professionals who find SCM patterns confusing and hard to apply.

Complex concepts require robust tools.

I teach one of the more complex CM tools because I feel that its power is well worth the investment of time and effort. I have written previously that I am not a big fan of lower end CM tools that lack proper security, audit controls and full (and secure) change history. I actually require that every developer in my firm attend a 2 day hands-on class (that I usually teach myself) in order to use our CM tools and I have just gotten back from a three week trip to India to teach and implement these technologies. So it would seem that I do sometimes accept that powerful technologies are worth the investment of time and energy to understand and apply properly. Are SCM Patterns worth the time and effort to understand and apply successfully?

CM is easy...

My boss used to say that CM was just a matter of knowing how to checkout and checkin your code. I felt very frustrated as I struggled to make him understand that there was a lot more to good CM practices than just the mechanics of a checkout/checkin. What are some of the real requirements of successful CM methodologies? Developers who work in large Global teams need the ability to organize their work in logical and manageable tasks. The

ability to isolate one's work from other people on the team and to integrate accurately and quickly is essential. This can certainly get very complicated very quickly.

OffShore and Variant Management

As I write this article I am waiting for one of my colleagues to call from India. Our task this evening is to work on merging the latest work completed by the New York developers onto the India development branch. The code is organized along a mainline (/main) with read/write access only in New York. Developers in India can see the mainline branch, but anything that they checkout is going to be checked out onto the India development branch (India_dev). Friday afternoon I reviewed, with the New York developers, the necessary steps to merge the code from the India development branch back onto the mainline. We have made things very complicated by deciding to allow OnShore developers the ability to modify the same body of code from both locations – at the same time (a lot of us do work in the middle of the night on a regular basis). Branching and merging is a bit advanced, but certainly within the abilities of most good software development professionals. Then a scenario hit me that I had not thought of!

There may be times when India based developers may need to do an (outer) merge from the mainline of development out to their own India development branch. This is necessary when the folks in India are changing the same piece of code that is then merged back onto main (by the New York folks) and modified further on the mainline again by the New York developers. This could cause the version that is on the India development branch to be out of date and require the latest changes from the mainline branch. How confusing!

SCM Patterns

This is exactly the sort of complexity that SCM Patterns has a great deal of power to manage successfully. The Patterns of Mainline, Active Development, Private Workspace, Integration Build, Repository and Task Level Commit provide the necessary tools to communicate these tasks and requirements successfully. The successful CM Practitioner will use the cognitive and software tools necessary to handle the complexity of the SCM task at hand.

Keep it Simple

One of the things that I really don't like about SCM Patterns is that I see a lot of places where things are more complex than (I believe) that they really need to be. So my position is still to keep things as simple as possible. For example, there are many references to using branches when Version Labels would suffice (assuming that your tool has the proper capabilities). I am only willing to adopt additional complexity when I absolutely must.

Worthwhile investment

I do believe that understanding SCM Patterns takes some effort. There are a of development managers who read these articles (and apply our best practices) who don't

Crossroads News

A Monthly Publication for
Software and CM Professionals

really have the time to learn anything more than is absolutely necessary. I often promote the notion of adopting Just Enough Process to get the job done. Yet for CM practitioners SCM patterns is a very worthwhile investment and will yield considerable benefits and value.

What if you have trouble getting into SCM patterns?

Well then you have really come to the right place. We are all professionals who do this work on a day to day basis. We teach CM tools and process, promote CM best practices and some of us literally spread these methodologies on a worldwide basis. Students who have been through my class, literally get to wake me up in the middle of the night if they have problems getting to their code! That means that I have a tremendous incentive to teach clear and practical CM concepts and practice. If I do a good job then I get a night's sleep! Read up on SCM Patterns and share your views (positive and negative!) on what works and what doesn't. At CM Crossroads, the Editors take great pride in presenting a variety of approaches and methodologies. We hope you will join us and participate in our Community!!

Bob Aiello is a Senior Editor for Crossroads News and an Associate Director at a major financial services firm in NYC, where he has company wide responsibility for Software Configuration and Release Management best practices. Bob is on the Steering Committee of the NYC Software Process Improvement Network (CitySPIN), where he is also the chair of the CM SIG which meets in Midtown NYC. Mr. Aiello has a Masters in Industrial Psychology from NYU and a BS in Computer Science from Hofstra. You can reach Mr. Aiello by email at raiello@acm.org

