

# Is your **way of working** driving you crazy?

Some **insights** into what works and the future.

A Q&A session with Ir Philippe Back, Way of Working Improvement Expert.

## **Q1: Modeling is said to be a good thing to do in projects. How is modeling useful in a project?**

To put everybody on the same page, one needs common ground upon which to share ideas, concepts, options, and consequences.

That's how modeling is useful. By creating some key models, the team can create a shared mental framework. I call that CCC for Core Concepts and Constructs. This applies to all disciplines, from the conceptual, down to the technical and physical.

As there are always too many details to deal with, simplifications are needed, and that is what models do provide.

## **Q2: What do you recommend as tools for modeling?**

I have quite a number of tools to recommend. One criteria that I do have is that each tool I select has to have an incredible cost/benefit ratio. Also, I want to have the cost of a tool staying in the hundreds of euros range. That way, a team can be equipped with the whole thing if needed without putting a huge dent in the budget.

Also, the tools have to be fast and efficient. I hate it when I do have a memory hog swallowing all of the memory of my system (like some applications in Java do for example).

So, here are a few I do like and use:

- Axure RP for user interface prototyping. It's a very good application that one person can grasp fast so that they can immediately be productive with it.
- Sparx EA for general modeling, including the whole UML series. What is nice about EA are:
  - the shared repository that does support true teamwork
  - the fast UI
  - the nice documentation capabilities, that are par to none
  - the ability to organize the work the way I need to and not some way that one thought would be for me
- Mind Manager for doing mind mapping
  - very good for capturing content during meetings with users, team members
  - very good layout facilities

- very good presentation mode for walking people through the content
- integration with lots of other applications on Windows
- Axon. Axon is a special beast, it is an "idea processor". MindManage works in terms of heuristic maps where Axon works with networks of thoughts.
  - it has a unique ability to prompt you with questions about the subject. e.g. how could you look at X differently, what makes X think you of, etc. Very very interesting to open perspectives
  - it can perform calculations and do weird things with concepts, something I like very much
  - all in all, an application I do like very much. On the technical side, this is one of the few apps that I do know of that have been written in Prolog.
- BBFlashback. Another must have for anyone having to explain stuff to people and share ideas.
  - it records your screen and actions
  - it allows you to convert the files to all kinds of movies, as in flash, mov, wmv, you name it.
  - you can add all kinds of balloons, zooms, sounds, accelerations,... to your movies to enhance them
  - and you add a soundtrack (or record it as you go)
  - it is very easy to use and a great webinar capture tool. Speaking of which, I do use it to record webinars I do attend on the web!
- Screenhunter Free. This one is just a screen capture tool. Like Snagit and so on. I am just used to it.
  - it has a nice crosshairs feature that allows very precise capture
  - it can save in all kinds of formats with my own naming convention
  - when I do have to capture the screens of existing systems to model them further, or to capture web page parts, this is very good.
- Jing/Screencast
  - Jing is a kind of screenhunter but with the added ability to add arrows and send things on the web in a private space.
  - Very good for adding URLs to entries in web tools!
  - But it is memory heavy and I have to run it only when I do need it.

### **Q3: People do work more and more in a distributed fashion. What do you use for supporting such things?**

- TikiWiki CMS groupware is my choice for maintaining the intranet/resource center
  - all features like file galleries, workspaces, wiki pages, faqs, polls, ...
  - its ability to integrate other tools is second to none in my view
  - the learning curve is quite steep but the benefits are enormous
- ProjectPier is kind of a dumbed down Basecamp
  - I do track all of my projects with it
  - the fact that is feature limited is in fact an asset.
    - I can use it on my blackberry on the move
    - it is fast and light
    - I can extend it if I need to.
  - people I do want to collaborate with grasp it within 10 minutes
- Mikogo - screen sharing service
  - free, works well
  - Now, skype integrates screen sharing but the resolution is not as good

- Instant Messaging, like Skype, MSN, IceChat, GoogleTalk are the tools of choice when discussing matters.
  - Skype is very nice with a good headset. I personally use I Microsoft thing that was given to me by my friend Luc Taesch and I am very pleased with it. It doesn't beat my other pair of headphones but they are good and light.

## **Q4: There is a lot of talk about Agile. What do you think about Agile?**

Well, there are a lot of things that are "Agile". Let's get back to the fundamentals of Agile. The Agile Manifesto is quite important in that it puts people over processes and tools for example. But this doesn't equate with a free licence to hack. .. over tools means that tools have to serve us and not become a chore to handle.

Agile is no religion for me. It is just something that came out of a need. This need I think was that complexity was overwhelming us and all the agile thing is to help us master that. Also, a lot of Agile leaders are people who have long internalized the things that do work and so, can perform well and elegantly.

An important thing related to model and Agile is to model with a purpose. That is agile. Be able to adapt yourself to the context. Not only have a hammer looking for nails!

Looking at the landscape there is Agile Modeling but there can also be Agile Project Management, we have Agile Data as well.

There is also Ivar Jacobson who moved to another concept, which is quite close. It is the Essentials side of things, as well as what he calls the Smarter way. I would say Smart with the "er" suffix.

Smart brings a sense of context. That's a bit what Alistair Cockburn had in mind with his Crystal set of methodologies. Be contextual. Do not be a zealot. Be a student of all and a disciple of no one. I do not give a damn about belonging to any church. What I do care about is results. Pragmatic, working, delivered results. That would be it about Agile I think.

## **Q5: What is the most common mistake you see when coaching people?**

A lot of people get into a hurry and jump into detailing a solution without having first thought about what the problem really is. That usually translates to a lot of money evaporating without any result. That's how you can end up having consumed 90% of your allocated budget and schedule and still only have a 5% in earned value. Quite a bad proposition where you end up in paying for working. There are lots of bad vibes occurring at these places. Better avoid this upfront. As you know, a project gets late by two years

one day at a time. That's also why Quality Assurance must be occurring throughout the lifecycle and not something people do not want to deal with. I am big fan of pragmatic QA.

## **Q6: Speaking of QA, there are lots of aspects to it. For example, Validation and Verification. What is the difference between the two?**

That's true. There are lots of various elements. As far as I am concerned, validation is user and stakeholder centric whereas Verification is all about checking that you have a system that is matching its specification. You can write a spec and have code that matches perfectly and still have users being pissed off. So, validation is not something you do at the end. It is something that starts on day one and goes with the system all over its development lifecycle.

## **Q7: When coaching teams to move up to the next level, what's the approach you recommend?**

It is important to adapt the interventions to the group. In a group, you can have early adopters, the early majority, the late majority, and the so-called "laggards", a term I do not like much. Maybe "jaded" is a better term. To get people to use the approaches, techniques, and tools, you want to have success stories. The best way to get them is to focus on early adopters first, and then moving to the early majority as soon as possible. Once you have a critical mass of people using the material and able to deliver with it for most initiatives, the late majority will take care of itself. The jaded people will then maybe "see the light." Also, I do recommend to work on real cases, not on classroom level examples. But doing that requires a lot of practice and is not for the faint of heart. I've got my share of, let's say, sweating moments.

## **Q8: What will one gain by investing in the approaches that you will coach them?**

First of all, they will gain clarity. Clarity of purpose, clarity of means, clarity of outcomes, clarity of responsibilities. All in all, the client will get a much increased efficiency. Obviously this is not going to happen overnight and there will be a need of sustained

investment. But the alternative is letting things going wild, foggy, and ultimately unprofitable.

You can obviously have profitability by tricking people, and using fog as a tactic to milk some more cash but this is not the way I want to work, nor the clients that I do want. I see a customer as requiring care from me so that they can grow to the next level. So, another thing clients will gain is an renewed sense of purpose and self-worth. Recognition is a pretty powerful currency when you look at it. That's what makes people thrive. Not money, nor big cars. That's what I help clients reach: a sense of achievement. And from there stems all the rest: more money, more margins, more satisfied customers.

## **Q9: What do you see as important trends these days when it comes to the way people do perform their work when dealing with software projects?**

The key trends I see are summarized by the following acronym: **CoDiTa MoWeb**

That's quite a mouthful. It stands for Collaborative Distributed Talent on the Mobile Web.

There is a massive trend that will smash us in the face in the coming years: the lack of available talent. A huge number of baby boomers are going to retire and this will leave us with a huge void in skills. It is already hard to find decent people on the market, and by decent I mean GOOD, and this will only get massively worse. The talent crisis is at our door.

So, to solve this, we will need answers. One are ways to handle CoDiTa, the Collaborative Distributed Talent.

Talent will get harnessed in a kind of swarm mode where people collaborate over the web and the key thing is that talent will have the choice to choose on what they do work. That's a completely different model than the one we see today in the industry.

The challenge will be: how do we get the swarm work efficiently? That's why I am investing a lot in shaping collaboration on the web for my own projects. I have done quite several projects with people I have never seen face to face, or even heard. All occurred through instant messaging and web pages. The delivery was outstanding tough. This also touches on e-reputation. That's the currency in the CoDiTa realm: trust and reputation. The come back of integrity and outsting of mediocrity. I like that.

The second part is the MoWeb, for Mobile Web. Everything is going to be mobile ultimately. My office stands in my briefcase these days, including CRM, phone, fax, groupware, messaging, development. It is the future. Just with the BlackBerry, I feel much more productive than with a laptop or desktop when on the go. An idea pops, and it gets captured in voice notes, get sent to someone to work on it. A project requests comes into the system, I do have to provide a GO|NO GO decision and that can be done in seconds without requiring me to be at my desk. I regained some social life with these mobile gizmos.

Twitter is the same, posting some daily insights from the device. Reviewing word and Excel docs is doable on it as well. And continued education on the move is perfect with the MP3 playback ability.

The mobile web is the future. And that future is here today.

So, CoDiTa MoWeb is my acronym for summarizing the important trends I do see. It is a mix of social and technological things.

## **Q10: I have a last question. I would like to know how you think the level of maturity will evolve in the coming years within software projects?**

That's an interesting question indeed. First of all, the level of maturity is very different from organization to organization. When working with software houses, who do products, the level is usually good. Even excellent. There most of the work revolves around standardizing the way of working.

When dealing with large companies, that's another story since the distribution of talent is really looking like a very flat Bell curve. You can have very good and educated personnel, working alongside people without a clue. That makes things quite difficult if the top management is not committed. That's why I do work with them first to define clear objectives to reach, a way to measure progress towards them, as well as an estimate of the value the initiative brings to the organization. This really needed if they want to succeed. We want commitment and not compliance. Commitment leads to continuing performance when the coach goes away. And that's what you want from a good coach: ending up as a self sustaining team and individual. A coach is helping you move towards the best you that you can be, and then let you flourish and reach new levels.

So, as far as I can see, I think that there will be a growing divide between teams that can deliver and others. Better be in the first group.