

FEATURED PAPER – NOVEMBER 2007

Why Agile Popped Up on the Radar When it Did

By Bas De Baar

In the first half of the nineties, I was studying Business Informatics at Vrije Universiteit in Amsterdam, The Netherlands. As a final exam, I performed field research at a large financial institution. The department in which my study took place was a pool of Project Managers, both novice and experienced alike. The projects they performed suffered from what they called “interventions”, which were changes triggered by the project environment. Being educated as a plan-driven-pro, I set out my checklist and searched for forgotten process components, only to find out after a couple of months that everything was neatly in place. From standard documents to procedures, they had it all. And still the project went from left to right.

Being the eager beaver that I was, I just kept on looking and looking for the missing ingredient but could not find a single clue to locating the feature or process that would help to solve the pressing problem. At one given moment, I had an “aha”-accident (a hit on the head), which turned out to be a life altering moment professionally. At the coffee corner, I overheard fellow project team members have a conversation about a procedure that they were not going to follow... My jaw dropped. Not following the official procedure? Not complying with company policies? If they didn't follow procedure then all the changes implemented were going to be completed without the project manager's knowledge... clonk. The penny dropped. It seems so simple now, but it really rocked my world at the time.

Obviously my research then took an entirely different direction; a direction that I am still following twelve years later by writing this text. My research started from an IT Project Management angle. I could almost find no clue to help me solve the problem in official literature. Of course there were some suggestions and ideas, but no final ready to use concepts that would take into account the behavior that I encountered at the financial institution. The best information I could find at that moment was Barry Boehm's *Theory W*¹. That specific text is still a big influence on my thinking to this very day. The point of this story is that, although I really searched for it during the first half of the nineties, I couldn't find antidotes to the problems of “plan-driven”. It might have existed, but it must have been perfectly hidden.

Fast-forward a couple of years to the second half of the nineties...

¹ Boehm, Barry W. and Rony Ross. “Theory-W Software Project Management Principles and Examples.” *IEEE Transactions on Software Engineering*. 15.7 (1989): 902-916.

The Second Half Of The Nineties

Enter Agile methods! These counteractions to plan-driven approaches are not the biggest methods and certainly not the most widely used, but they are mainstream at this moment. They popped up on the radar somewhere at the end of the last millennium. What had happened? Why did they come into existence at that moment? Why not sooner? The problems were there. The knowledge was there. Why did we see agile around 1995-1999?

There must have been industry complaints about the plan-driven approaches. If everything was fine and dandy, nobody would have bothered to invent something new. There were, and still are, fundamental ones².

Problem 1: Management is thinking alone about how the job has to be done with little or no employee input (central planning). In as complex an environment as a software project, it is impossible for non-experts to know what the exact planning should be. Of course, the project manager will be the negotiator that integrates all different aspects, but it is a team effort overall.

Problem 2: Management is thinking that the paper plan will automatically be executed in the way it is described. People are not machines. You cannot throw instructions to the bottom of the corporate food chain, expecting that the employees will pick it up and execute the instruction without question. This is evidently not going to happen. Individuals need motivation. They need to be involved in the creation of the task they will have to perform. I once wrote: "*How to ruin a perfectly good process? Give an order.*"³ With the highly educated people we have in the software business, it is inevitable that they have a strong will of participation when it comes to determining what they should do and how they should do it. Also, in regards the time frame in which they should complete a task, they have a strong opinion. Not allowing them to participate in the "planning" stage thus reduces the commitment they have for the work to be done. Elements of thinking such as "not-invented-here syndrome" and "if you think you know better..." will ruin the project, which is directly linked to problem one.

Problem 3: Management is thinking the plan is 100% correct (there is no room for deviations). With all the uncertainties that are present in a real project, there is no way you can predict the future in advance. The plan will not reflect the reality just because of the simple fact that the plan is wrong. This is no big deal; this is just as it is. If you

² <http://www.cpgec.ufrgs.br/norie/iglc10/papers/47-Koskela&Howell.pdf>

³ "Surprise! Now You're A Software Project Manager", Bas de Baar, Mutli-Media Inc, 2006

try to keep on changing the process, to steer reality towards the preset expectations, you are doomed.

A side effect of this is that people are judged by how well they comply to the preset goals in the plan most of the time. If you do not comply with the goals, this is seen as a failure. The creative solution to counter this is just to report that you met the goals, even if you didn't. A few years ago I worked under chief of a project management office that always changed the progress reports in such a way that the overall performance of the PM office looked good. The report used smileys (yuk) to indicate any risks. Red smileys signalled something potential not under control, and you can have none of that! Even if you don't manipulate the indicators, you would still use a large amount of your time explaining why you didn't comply with the preset plan. The real reason (the plan sucked) is not considered a sufficient answer.

Another interesting point that Lauri Koskela and Greg Howell⁴ raise is the lack of information about how the project plan should actually be performed. Management creates a plan, throws it to the employees and provide no comments on how they should operate. We all know that to start a specific task, some of the necessary materials or resources might be missing. Then the informal and creative process of fixing and improvisation will start to allow to evolution of the task anyway, or have a direct impact that minimizes the effects on other activities. Although the execution of project tasks generally consists of improvising to make the scheduled tasks happen, the plan driven methods provide no underlying basics for the execution. There is just the plan, and if something has to be changed, it alters the entire plan itself.

The Tipping Point – ISO 9001

However, being bad is not enough. As explained earlier, the forces that pushed planned-driven to the acceptance phase were (and still are) very strong. It was necessary to create momentum for people screaming for an antidote. In my opinion, the ISO 9000 craze finally pushed us all over the edge. The idea seems simple - you describe what you do and perform what is described. Independent people would check if your organization had the documents in place and followed them. The idea behind this "certification" was that everyone outside the company would be guaranteed that you produced high quality products. Sound familiar? It smells like Scientific Management. Actually, it should have smelled like the continuous improvement and the systems thinking approach that had made Japanese production so famous. Their car industry had a great reputation with the way quality was handled. It was moved to

⁴ <http://www.cpgec.ufrgs.br/norie/iglc10/papers/47-Koskela&Howell.pdf>

Europe and North America, where we turned it into an industry of compliance and auditing, Taylor-style⁵.

In 1987, the notorious standard was adopted by the International Standards Organization after initially being introduced into the UK. It was intended to bring Japanese perfection to the West. It was initiated by governments (the United Kingdom primarily) and had quite a force behind its propagation. Companies that wanted to do business with governmental organizations had to be ISO certified. Companies that worked with them were also required to have an ISO stamp of approval. Its spread took epidemic proportions. The beauty of the ISO scheme is that it influenced everybody within the affected organizations, from the doorman to the board member, from the secretary to the maintenance guy. The group, swamped under the enforced bureaucracy, got larger and larger. It didn't take that long for the people to realize that the documentation had nothing to do with quality, but was only present for compliance. The secretary and the maintenance guy, they all got a piece of the "create documents just for the sake of documents" paradigm.

Agile Manifesto

More and more people were touched by the bureaucracy and control structure that seemed to exist for their own sake with enough momentum to preach an antidote without being immediately treated like an infidel. The dawn of Agile. It is not hard to see that it is a direct counter-reaction to the plan-driven approaches. A group of people working on methods around software development, formulated their believes in the Agile Manifesto:

"1) Individuals and interactions over processes and tools, 2) Working software over comprehensive documentation, 3) Customer collaboration over contract negotiation, 4) Responding to change over following a plan"⁶.

Away from heavy duty prescriptive procedures, away from products just for compliance, human interactions to the center and an alternative to the idea that the plan defines reality. It was presented as everything that plan-driven is not.

Two other things happened around the same time that moved non-traditional project approaches out of their isolation. Both were triggered by the appearance of the World Wide Web. The Web made it easier to spread the word about the new approaches. It was easy to publish, low cost and had a very good audience, especially during the early

⁵ <http://www.lean-service.com/3-1-article.asp>

⁶ <http://www.agilemanifesto.org/>

days of the software industry. Previous publications about project management had to go through peer review, which triggered a kind of convergence of views. For entirely different approaches, it was harder to get publicised. The Internet also created a better way to find the information one needed as a result of search engines. The Internet technology created new business opportunities: business got faster, business got even more demanding. Emerging was a need for methods of doing projects faster. If there is money to be made, business people are open for any type of new ideas.

Agile appeared during the mid-nineties because ISO 9001 had created a change receptive crowd. The Internet helped to spread the word quickly and provided the business with incentives to be open to the suggestion.



Bas De Baar
Author



Bas De Baar works as a Project Manager within the publishing industry. Since 2001, he has been the editor of www.SoftwareProjects.org, a popular website dedicated to Software Project Management. He holds a masters degree in Business Informatics and lives with his wife in the coastal town of Zandvoort, The Netherlands. His book, "*Surprise! Now You're a Software Project Manager*", was published in September 2006 and became an international best seller. It is available from www.SoftwareProjects.org/surprise.htm or from most book retailers. Mr. De Baar can be reached at basdebaar@gmail.com.