

Agile Thinkers

A blog about the many great thinkers in the Agile community. Hosted by Clarke Ching

May 11, 2008

Tom Gilb - an example from Jens ...

Jens Egil Evensen has been using Tom and Kai Gilb's EVO approach with Scrum. Here's a few questions and some very interesting answers with Jens. I strong recommend you read on ...

A very

Q: Hi Jens, Tom said that you've been using Scrum combined with EVO. Can you tell me a little background about the project and what problem EVO solved that Scrum didn't?

I've been studying and using Evo for the past four years. First as an employee in the Norwegian Product Register (governmental) after attending a course with Tom Gilb,, and later as a consultant in one of Norways largest IT corporations (EDB ASA (Avenir)). Avenir has a rather complex and "heavy" framework for managing large waterfall like projects called AvePro ("homemade" PMI inspired) that they have been working on and used for 20 years.

One of my first assignments in Avenir (in December 2006) was when Microsoft asked us to be an implementation partner for a Norwegian telecommunications company that had decided to implement Microsoft CRM 3.0 as their standard CRM platform. The requirement specifications that where presented to us where purely functional, and there where no references to what goals or business values they wanted to achieve by implement the system. I quickly concluded that our traditional methods of project management would not cope with such a project. One of the reasons where that I suspected that if we just developed a system based on the specification that we had received, the changes in requirements during the course of the project would turn the project into one giant contract nightmare, and that delivery of the system would be infinitely delayed. It would simply join the majority of all other software development projects in the basket of failure.

We agreed to do a one month test period with one consultant (me), kind of no cure no pay, to qualify us and the system for their needs. I told the customer that with the method I intended to use, we were quite sure to give at least some stakeholder some value, and that this value would be far beyond the cost of the initial implementation of the project.

To do this I had two simple requests for the customer. I wanted one workshop with the sales managers ad the director of sales, and one workshop with the employees from their internal IT department that where suppose to participate in the project.

I started the workshop with the management by asking two simple questions:

- 1: What is this company's vision?
- 2: If you don't think about IT or software at all, what are your greatest problem today, that prevents you from fulfilling this vision or goal?

(BTW, I'm not of the very formal types, so I attended this meeting with the management in my usual work outfit; T-shirt and jeans – in Norway that is almost perfectly okay)

At question #1, they answered a bit staggering, and after a short while, they said with a little laugh that is was their vision to be a leading telecommunications company in Norway. Laughing as if they where embarrassed by this vision (I call this Norwegian modesty). I thought that was an excellent vision, and a good starting point for our project. No point in setting the goal to low. If you're aiming for the moon you are more likely to reach the sky than if you only aim for the roof of the nearest building.

At question #2, the director of sales had a clearer view. He said, and I quote: "We are loosing xx million Norwegian kroner each year on contracts that are ending and lost, without even trying to win them back, or prolonging them. Nobody knows they even exist".

I answered: Okay, so if I implement a function in your system that alerts the account manager of a customer, let's say, 2 month's in advance that a contract is ending, you will potentially earn xx million kroner more?". He answered yes.

This was actually a function that they could have implemented in their existing system at any time, but the management of sales had never discussed this business related problem with the IT department, and the IT department where probably to busy surviving bugs in the existing systems to listen to the real stakeholders everyday problems.

I told them that I could implement such a function for them in only 14 days and where met with a short silence... Then the director of marketing said: I never thought I would hear such a thing a IT consultant".

In the workshop with the resources from the IT department, we discussed all the practical tasks necessary to install and configure the CRM database and server, and some other technical stuff. Just enough for me to gain their respect, and to make them know that I really knew what I was talking about when it came to programming and technical issues.

I also told them that to be real heroes to the management, they had to deliver solutions that really made some increased business value for important stakeholders in the management. Cool features might be impressive, but they are short lived and soon forgotten. Business value is sustainable. They where very skilled resources, and had many years of experience, but they had never thought of it in that way.

I told them: When your boss ask you to do something for him, you must ask why. Only then will you know what his real goal is, and only then will you be able to really help him. The answer to his question is never yes or no, it is rather one or more design ideas with an explanation, a cost estimate, and an impact estimate on his goal and other relevant impacts (good or bad). This information will make your boss able to make a smart decision on whether the solution should be implemented or not, and it will make the IT guys real heroes in the eyes of the management.

After working in this way for a month, creating business value they really needed, and showing the potential of the method, they prolonged the contract, and wanted more resources from our company. The total number of hours for this project became about 5,300. In Norway, that is considered a mid size to large project. We kept delivering business value to the customer in small weekly (!) increments for 15 months.

When the number of people working on the project increased, organizing the work became challenging. Mostly because of the rapid increments in the project, and that not all of the developers working on the project where experienced, and needed more follow up. Evo is a great method to break down top level goals into business value and end state requirements, but contains no mechanism to organize the developers, and how they work. Although design specifications where made in detail, we had trouble with developers that where developing outside the scope of the project, and struggling for days with problems that weren't important to solve at all, without telling anybody (also called thrashing?).

To get the necessary control of the production environment Scrum where partly implemented with one team, daily stand-up and a product backlog. That product backlog where populated with design specifications from the Evo cycles. This, I found, was an ideal combination. Now I had total control of the most important requirements, expressed as the business value they wanted to achieve. A breakdown into end state stakeholder requirements. Mapping between business value and designs that where implemented. The stakeholders received business value for their money (bang for bucks), and the programmers no longer wrote code; they created business value!! Their work had suddenly got meaning, and they loved it.

As far as my knowledge goes, no single method widely used today covers all aspects of a project, from Vision to values, values to requirements, requirements to design, and design to working systems.

All methods seems to cover the "project layer" where the author of the method works, and maybe one layer up and one layer down. To cover all layers you will have to use more than one method, and select methods that have compatible interfaces, such as Evo and Scrum (Evo design specifications into the Scrum product backlog).

Avenir has now added this methodology (in a modified form) as one of our preferred agile project delivery methods (Avegility). It's built up as three parallel cycles named "Goals and requirements", "Production (ie Scrum)" and "Delivery and measurement".

Q: How did you learn enough about EVO to get started? Was that easy?

As mentioned earlier, I first learned about Evo when I attended a course held by Tom Gilb, and I must admit that the first two days of the course, I did not understand where he was going with this at all. After the course I tried it in a project at work, and I was truly amazed with the result. Lifting my eyes from the code and the software, and focusing on the required results opened up a whole new world of possible solution for me. As a consultant for a software company I must admit that this sometimes is a problem; when looking at clients requirements and goals today, I far too often sees that the best solution seldom is buying or developing new software, but to begin with the existing software, or the way they work.

If you got that business gene, and a true passion for the results you create, living the Evo is easy. If code, technicalities and gadgets are all you live for, you will probably not get the point at all.

Some people never realize that the best system is the one delivering the best results for the business, and not the fastest, prettiest and the one with the perfect architecture (although we have to care about those things as well, but then as expressed quality and performance requirements related to business values and goals)

Q: Did you have to change Scrum much?

I suppose I get a lot of people angry now, but in this particular project, because of the short cycles, we removed the Sprint term, and worked continuously with developing and delivering items from the back log, meeting every day in the daily stand-up. In the formal method Avegility Scrum is unchanged but for one thing: It's not the product owner that is responsible for handling and prioritizing the product backlog, it's the Evo cycle's output in the form of design specifications. Cost/benefit ratio decides the priorities.

Posted at 02:07 PM in [Tom Gilb](#) | [Permalink](#)

TrackBack

TrackBack URL for this entry:

<http://www.typepad.com/t/trackback/12757/28975252>

Listed below are links to weblogs that reference [Tom Gilb - an example from Jens ...](#):

Comments