

# From CMMI and isolation to Scrum, Agile, Lean and collaboration

Mads Troels Hansen,  
BestBrains  
mth@bestbrains.dk

Hans Baggesen,  
hansbaggesen@gmail.com

## Abstract

*This paper describes a journey from 2004 to 2008 when SoftwarePeople in Denmark, together with a partner from Bangladesh, established a subsidiary company more than 7000 km away from Denmark. We hired 20 people in one week in Bangladesh and started to use CMMI processes to integrate development teams between the two locations with the goal of receiving a CMMI level 3 certification in 1.5 years. After some challenging time we stopped the CMMI project and switched back to Agile and Lean techniques with more collaboration. Here we describe our experience with implementing global big bang Scrum and building a kaizen culture. A journey from long running projects, technical dept and integration nightmares to small batches of work, continuous integration and faster delivery of business value. This is reported by Hans Baggesen, Team Lead for one of the Danish R&D teams (2007-2008) and Mads Troels Hansen, CTO and co-founder (2004-2007).*

To learn from other companies working with different offshore setups, we contacted both software companies and production companies. We also talked to different consulting companies with expertise in distributed development to find out how we could make our offshore setup more productive and successful.

The different experts and experienced companies stressed how important it was to establish shared processes for everyone to use. The different process consulting experts told us the practice to use in a offshore setup, was implementation of CMMI processes on both locations. Initially we made a feasibility study for starting a new company in Bangladesh and implementing CMMI processes with help from an external consulting company. It looked very promising, even though we could see a lot of challenges.

## 1. Why a company in Bangladesh?

In 2004 SoftwarePeople had developed some innovative and advanced software to establish the technical infrastructure between Denmark (DK) and Bangladesh (BD). The infrastructure connected our sister company, AdPeople, with their new graphical production subsidiary in Bangladesh, GraphicPeople. The company in Bangladesh was founded together with a Bangladeshi partner. At the same time, it was not easy to hire skilled developers in Denmark, so in SoftwarePeople we decided to look at different options to scale the development department outside Denmark. Our motivations were cost savings and access to competencies. After visiting several software companies in Bangladesh in 2004, we decided to start a subsidiary software company with our current Bangladeshi partner.



Figure 1: The Danish and Bangladeshi office

We decided to seek help with CMMI process from Indian companies, because they had experience with CMMI and were not as expensive as the Danish consulting companies. We started looking at Indian CMMI process companies and selected 6 companies for further study. In February 2005, the Danish management team went to India and visited the selected companies. We found one promising partner and signed a contract with them to help us reach a CMMI level 3 certification in 1.5 years.

### 1.1. Hiring people in Bangladesh

Together with our local partner in Bangladesh, we created a couple of full page job ads to the main newspapers in Dhaka (the capital of Bangladesh). We were looking for many different roles for the new company (management, QAs, project managers, developers etc.). We got a vast number of applications and our partner in Bangladesh did the first selection of candidates based on different criteria and also did the first round of interviews.

To start up the company, one of the very experienced directors from the Danish organization was relocated to Bangladesh for two years as CEO. In July 2005 the Danish CEO, CTO and the future

Bangladeshi CEO went to Bangladesh for the second round of job interviews with about 100 people in one week. Our partner had selected the candidates for this second round. It was a fantastic, very interesting, but extremely hard week talking with so many different people. We found enough talented and skilled people for the different roles, to be able to establish 2-3 teams from the beginning.

## 1.2. One month in Denmark

All the new people from SoftwarePeople Bangladesh went to Denmark for one month in the autumn of 2005 to work together with the Danish teams, understand more of the business domain and establish a better cultural understanding between all the teams.

From the beginning of 2005, we had been working together with Microsoft on the TAP (Technical Adoption Program) for Team Foundation Server 2005, and after a lot of experiments with the different beta versions during 2005, we went into production with TFS (Team Foundation Server) beta3 as one global system for teams in both locations. One place for all source code, documents and builds. We managed requirements as work items in TFS with easy access for everyone. We did some training on how to work with TFS for the different teams and simulated distributed scenarios with all teams co-located in Denmark.

## 2. The first 6 months with CMMI focus

We had a strong focus on the processes between the two locations and therefore the goal was to reach a CMMI level 3 certification in 1.5 years time. At that time we were not aware of the impact of defining a certification as a goal to integrate the two locations. Instead of having a goal to create business value and deliver efficient software solutions to our customers, we had a strategic goal to establish a process framework and fulfill a number of CMMI KPA (Key Process Areas).

To start our CMMI journey, we worked with a very skilled CMMI consultant from our Indian partner, and he did some formal CMMI training for both the Danish and Bangladeshi teams. The CMMI consultant also did a GAP<sup>1</sup> analysis with a light CMMI level 3 assessment of our current processes. One of our skillful project managers was appointed project manager for the CMMI project. The challenge was to close the gaps identified in the gap analysis. We established process

teams with people from the different development teams to create templates and describe processes for the different areas related to all the CMMI KPA required for a level 3 certification. A lot of interesting discussions were conducted in the different teams and we tried to envision what our future “perfect” processes would look like.

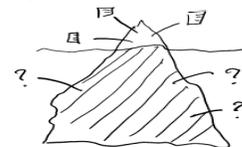
The process teams were very energetic, but it was also obvious that we started focusing more and more on producing documents to fulfill CMMI KPAs for CMMI level 3, rather than delivering good business value to our customers. Our total cycle time for work request was increasing and we started to communicate more and more in documents with our customers and between people on the different teams in both Denmark and Bangladesh.

After discussions with some of our customers, the people working with the process teams and in the management team, it was evident, that:

- None of our current customers requested that we had the CMMI certification
- It looked like the focus on communicating in documents between teams isolated them from each other
- There was a starting culture of “us” and “them” with misunderstandings of the written requirements

Was this just a temporary condition? Would it be better when the CMMI process platform was established? Looking at our different problems, we found two main areas related to the structure and understanding of the work:

- We created requirement documents for solutions to a problem domain, which was not understood by teams in both Denmark and Bangladesh
- It seemed we documented requirements for the part of the iceberg above the surface without considering the part below it.



**Figure 2: Focus on the visible part of the iceberg**

A third problem was the increasing time used on integrating code from different teams. We worked together with Microsoft Redmond on the best practice using TFS with distributed teams, and they recommended a source code structure with a branch for each team and one main branch for the product. In the past we had used Continuous Integration for our source

<sup>1</sup> GAP analysis: [http://en.wikipedia.org/wiki/Gap\\_analysis](http://en.wikipedia.org/wiki/Gap_analysis)

code with one mainline for all new development. When we started to work with different branches for each team, we spent a lot of time doing merging and resolving conflicts. It was one more reason to blame other teams, when “their” stuff did not work after a merge.

We needed to improve:

- Collaboration and communication
- Technical and business feedback loops
- Cultural understanding
- Domain knowledge and requirements

### 3. Implementing Scrum in two locations

In the spring of 2006 we started searching for alternatives to the heavy CMMI processes and looked at experience with Scrum in a distributed setup. It looked interesting, and we decided to find out if Scrum and Agile practices might be a way to increase productivity and collaboration between the distributed teams. As a result of that, the CTO and 3 project managers took Scrum Master training in June 2006. At the end of that month the two CEOs from Denmark and Bangladesh together with the CTO, received Scrum Product Owner training in the UK to establish a shared understanding on Scrum. At the same time we started experimenting with Scrum in one of the Danish teams working on our most complex project. The results were so good, that we made a global Scrum rollout. We started with local teams in Denmark and Bangladesh and decided to put the CMMI work on hold until Scrum was up and running.

The Scrum implementation strategy was to start with local Scrum teams in both Denmark and Bangladesh working on R&D or customer projects (figure 3). When they had established a stable rhythm in the local teams, we would move into global teams with team members from both Denmark and Bangladesh. One of the Danish Certified Scrum Trainers (CST) made a Scrum introduction to the whole Danish organization and a couple of people from Bangladesh who worked from the Danish office at that time.

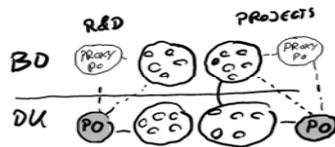


Figure 3: Initial Scrum team structure

We started a couple of training programs in both Denmark and onsite in Bangladesh. Most of the sessions were conducted by the CTO, and it made the

management focus on the Agile transition visible for everyone. We scheduled training sessions in areas like:

- Scrum roles, backlogs and rhythms
- Requirements with user stories
- Estimation with planning poker [2]
- Retrospectives [1]
- TDD, automated builds and CI
- Task boards

All the Scrum teams worked with physical task boards in their team room to manage the requirements and tasks for their sprints (figure 4). The requirements were linked to TFS work items.

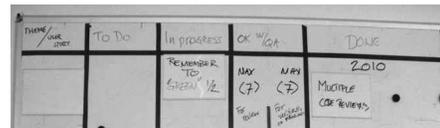


Figure 4: Task board from one of the local teams

After a few sprints, we saw that productivity had increased substantially and that the different teams worked with renewed energy. There were also a lot of questions and concerns on what to do and what not to do. To embrace and implement Scrum across the different teams and organizations, we focused on doing regular retrospectives on both team and company level. The first couple of company retrospectives were quite hectic and required some extra facilitation techniques to be constructive. But we did identify some of the main problems when crossing the team boundaries and working together with other departments.

Initially we started with the Product Owner (PO) as a role for sales and account management. It worked reasonably well with local teams in Denmark when they could collaborate and meet when required. But it did not work well with teams in Bangladesh. We moved some of the very experienced senior developers and consultants into global PO roles to establish a more structured flow of tasks and collaboration with the teams.

To establish a structure to improve the flow of work between the Danish and the Bangladeshi teams we introduced a “Proxy PO” for each team in Bangladesh (figure 3). We also started to mature and make requirements ready for the teams to pull into the sprints. In TFS we changed our Scrum process template to include “accept criteria” for each user story.

Understanding the problem domain and how the different requirements were related was still hard. We used many different techniques to move the domain

knowledge and user understanding out to the different teams. We had several domain knowledge sessions in Bangladesh with the theme “what you don’t know, you don’t know”. In those sessions we created mind maps of our product, to visualize relations between modules, external systems, workflows etc. We also used physical material created by customers using our product to visualize why and how the customers would use the product.

### 3.1. Global management system

To establish a structure with a regular coordination rhythm in teams and between teams and management, we established the following SoS (Scrum of Scrums) structure with four levels of daily meetings (15 minutes each):

- **Scrum meeting** in the team
- **Developer Scrum of Scrums** between teams working on the same product/project
- **Scrum of Scrums** between Team Leads (Scrum Masters)
- **Meta Scrum** between Team Leads and CTO

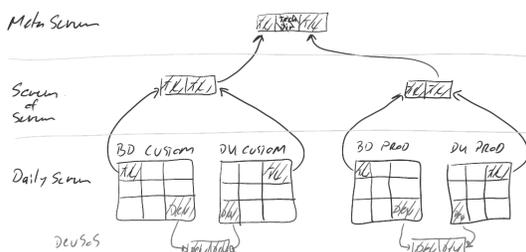


Figure 5: SoS structure in 2006/2007.

The SoS structure enables fast escalation of problems from the teams into the SoS meetings and Meta Scrum. We realized that the developer SoS meetings did not work that well because each team was more focused on their own sprint commitment. Helping other teams with their tasks and commitments was a second priority.

### 3.2. Retrospective: Product focus still missing

No matter how much communication we tried to establish between the teams, we were still far away from shared team focus on the complete product. As a result we could see the architecture becoming more fragile, multiple code dialects and distrust between teams were also starting to grow.

By doing regular retrospectives with all the different teams and across the company we identified the main problems and decided it was time to create the global Scrum teams. We wanted to improve

communication and trust by working on shared commitments between the different teams.

## 4. From local Scrum to global Scrum

The following section is about how the local Scrum teams became global Scrum teams with people from both Denmark and Bangladesh (figure 7). The primary focus will be on the teams working with R&D.

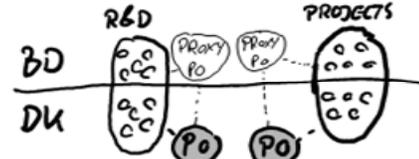


Figure 7: Global Scrum team structure

### 4.1. Global R&D Scrum team

We still needed to improve the focus on the complete product and not on "bad code" from other teams. The main barrier for building quality into the product was the distrust between the teams who worked on the same product. Therefore the two R&D teams in Denmark and Bangladesh were merged into one global team. Instead of two distributed teams, working with different sprint backlogs, we established one team with one sprint backlog. The problem of building quality into the complete product was now a shared team problem, not a management problem.

### 4.2. Global planning

Having one team and therefore one sprint backlog meant, that we had to do sprint planning together. Not an easy task, but after some experiments, we came up with a solution that worked for us:

- The whole global team was presented with the user stories online
- The global team went offline and started to break down the stories and meet to synchronize sprint backlog items into one sprint backlog
- The global team informed the PO of the result

We spent a lot of time trying to make it work well, and it required a high level of trust between all parties. It also required a lot of knowledge of the business domain, but made it easy to ask for the relevant knowledge when doing the breakdown and planning.

### 4.3. One global shared focus

Being one team and pulling tasks from the same sprint backlog, we were able to talk about the daily work with the other global team members instead of

waiting until the end of the sprint. The daily talk in the team evolved around the shared unsolved problems and not around errors in the past and who was to blame. The daily dialogues also created more trust between the distributed team members, because they worked on the same commitment and shared goal. As the team progressed, we came to do sprint within sprints, breaking each user story into several minor user stories that each brought some business value. These minor stories were built on one of the locations, but the entire story was developed by the whole team. The team could now deliver a quality feature, from idea to production, in between 5 and 10 workdays. The communication was improved on many different levels. Daily Scrum meetings and global code reviews created the extra trust to move business value much faster with better quality.

#### 4.4. The daily global Scrum meeting

The daily Scrum meeting on a global scale was an easy move, because it consisted of two things: 1) When to do it and 2) Converting our physical task boards into a virtual task board. One of the main cultural problems with the daily Scrum meeting, “Reporting to Scrum Master”, instead of synchronizing knowledge between colleagues, had already been removed. Our physical task boards were replaced by a web application (figure 8). It took a senior developer a day to build a first version of this virtual task board around the TFS API. In the following 5 sprints more features went into the virtual task board.

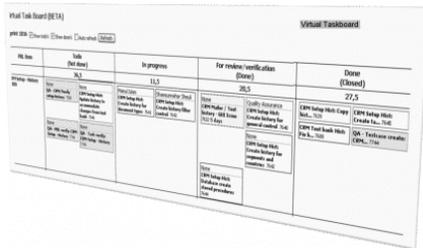


Figure 8: Virtual task board integration into TFS

#### 4.5. Global code review

The global code review – was the major mover in building trust and improved feature completeness. When a task was done and could compile on the local developer box, the developer would request a global code review. Code built in Denmark was inspected by a team mate in Bangladesh and vice versa. Looking beyond building better relations in the team, the economics involved was interesting as well. There were fewer errors slipping through to the QA reviews, on average reducing the time spent on fixing errors in

the sprint by 50%. Automated testing improved this further.

#### 4.6. Moving people around

Code review is difficult to do; it often results in one of the parties feeling insulted. So in order to do code review we needed to have the initial level of trust established between the developers. Both code review and the daily stand up meetings are just a burst of professional communication, while moving people around builds up the social relations between the team members. So we started to move people around on a steady interval between the office in Denmark and the office on Bangladesh [3]. This was not an easy task because we needed to acknowledge different cultures’ needs and family structures to make it work. In our globalized world it is amazing that cultural understanding is not mandatory when taking a computer science or business manager class.

Moving people around made code review and daily Scrum meetings much smoother and therefore made building code easier and more fun.

#### 4.7. PO collaboration with the team

We established the trust on the global team, the quality of the code raised, but there was still not a high level of trust between the developers and the PO. The team working with R&D, were very dependent on the PO and his ability to share the vision and the market focus with them. The teams working with customer projects had more specific customer requests and problems to understand and implement.

The R&D team could build the best quality code, but we kept building elements of the wrong things. It was almost like building a top of the line V12 engine, for a solar cell car race. The trust between the team and the PO were at stake. “I told you to build an engine, and you should have known it was an electric engine I needed”. So once again we had to improve the quality and understanding of requirements between the PO and the global team both before and during the actual development in the sprint. After retrospectives and more discussions, we decided to try moving the PO into the Danish R&D team room.

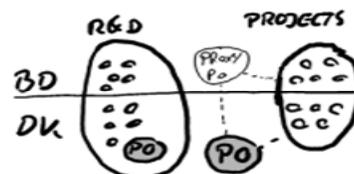


Figure 6: Later global Scrum team structure

By moving the PO into the same room as the developers, we created more frequent access to domain and business knowledge for team members in both Denmark and Bangladesh. But along with the PO working in the same room as the team came feature creep and different chatter disturbances during a sprint. To manage those problems, the PO became a team member of the global R&D team (figure 6). The PO was now part of the daily Scrum meeting, team commitment to complete a sprint and retrospective. This moved all of the work needed for bringing a business opportunity to completeness, into a more efficient system with more collaboration and understanding. Still we had focus on pull processes to establish focus and rhythm for the team. PO activities like planning the roadmap and grooming the product backlog became more visible on the team radar, and they now could help the PO to make better decisions. The global team felt more responsibility for the whole product and they not only developed the things the right way, but increased the focus on developing the right things. The domain understanding and collaboration was improved a lot between everyone.

## 5. Key learnings on this journey

We found that the focus on CMMI processes and having a specific CMMI certification level as a goal isolated the teams from each other and created distrust. Moving into Agile and using Scrum engaged people in much more collaboration and visualized many hidden problems. Working with the problems and doing regular retrospectives, created more trust and understanding between the distributed teams. It also changed the focus from having the processes as a goal to having the process as a tool to deliver maximum business value with high productivity as a goal.

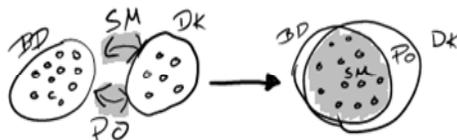


Figure 7: Into a global Agile environment

To be able to work with Agile practices in a distributed setup, we had to establish and focus on specific areas to manage the physical distance between the team members. We found that especially the following 8 areas created a global Agile environment (figure 7) for the people to build the necessary trust, collaboration and understanding to continuously improve their daily work and build great software:

1. Global structure, to manage the work with value streams, focus and boundaries

2. Establishing technical and business feedback rhythms
3. One global technical infrastructure
4. Continuously improving development practices
5. Requirements management and Agile planning
6. Training the teams in knowing the problem domain
7. Cross culture understanding with more collaboration and onsite visits
8. Communication protocols

The organization was going through different phases in this journey (figure 8). Starting with CMMI and more prescriptive processes (1) and then moving onto Scrum and adaptive processes (2). Later the local Scrum with distributed teams became more prescriptive and static (3). With Global Scrum teams we again moved into more adaptive processes and improved collaboration (4).

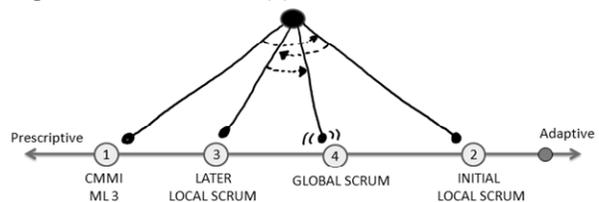


Figure 8: The process journey

## 6. Where is the company today?

In a rather short time frame, SoftwarePeople had moved into a mature and high efficient company with distributed teams delivering high business value. In the year 2008 they were acquired by a big US corporation and they are now part of a global network of companies working with marketing for some of the largest companies in the world. They develop and support global software solutions in EMEA, Asia and the US with special focus on the US. It will be an interesting journey to adapt to this new and demanding setup with people and customers in almost all time zones around the world. The CMMI work is still on hold.

## 7. References

- [1] Agile Retrospectives, Diana Larsen et alt., Pragmatic Bookshelf, 2006.
- [2] Agile Estimation and Planning, Mike Cohn, Prentice Hall, 2005.
- [3] Xtreme Programming: Move People Around [Online] (Last modified: Unknown). Available at: <http://www.extremeprogramming.org/rules/movepeople.html> [Accessed May 27th 2009].