

# Scaling Scrum @ Thales with Emergent Innovation

Cesario Ramos & Sandra Roijakers

## About Thales

Thales' strategic vision is "Working together for a safer world". Whenever critical decisions need to be made, Thales has a role to play. In all its markets – aerospace, space, ground transportation, defense and security – Thales solutions help customers to make the right decisions at the right time and act accordingly.

One of the business lines in The Netherlands is Surface Radar. Within Surface Radar is the Technical Unit Processing (TU Processing), responsible for the development of the digital processing functions in radar and optronic sensors for naval and air defense applications. Multidisciplinary teams pursue these developments, with specialists in the areas of system engineering, sensor functional design, architecture, design and realisation of real-time embedded systems.

## Start of the agile era

About ten years ago TU Processing, a 70+ people department, heard about the existence of an agile manifesto and the Scrum framework. There were doubts whether this would fit to the situation TU Processing has to operate in: a high tech embedded software development environment with defense requirements on the way of working, and without contact with the end customer. This is quite different from the user interface world where most agile success stories came from. However, the promised status visibility and intrinsic knowledge sharing sounded very attractive. Hence, the department started to experiment with agile. First with student projects, then research projects. Later on two dedicated small development projects in the radar domain. The promises became true, and the involved engineers were positive. Hence it was decided to take a next step.

## Change is needed

Most projects in TU Processing run for multiple years with tens of thousands of man-hours needed to complete them. The Processing domain consists of several very different subdomains, varying from sensor management to waveform calculation and burst management, and to the subdomains of the radar processing chain where the raw data is stepwise converted into an environmental picture with clear tracks. Each of these subdomains requires dedicated functional knowledge and experience, and working up into a subdomain takes months to years.

Over the years the expectations of TU Processing increased steadily. To cope with the challenges of today's market, projects need to be delivered faster, without compromising reliability and quality. In order to deal with these challenges, Peter – as head of TU Processing - decided to roll out Scrum across the whole department.

### *Success through increased transparency, -control and -focus.*

The main objective was to remain a reliable partner for the stakeholders, being the projects. The department was convinced that the increased transparency, control and focus on value would enable it to improve its development success.

## **Scrum and ...**

Sandra is a cluster manager at Thales and is a member of Peter's management team. She works with a number of Scrum teams and takes care of the daily operations and impediments inside the department. Together with the engineers that were part of the first Scrum teams, she started as the evangelist of Scrum and agile development and spearheaded the Scrum adoption. Knowing that stable cross-functional teams would be a great start, the first big change was implemented: the move from project teams to stable long lasting cross-functional Scrum teams.

### *Move from project teams to long lasting stable teams.*

This change meant that:

- Instead of resourcing dedicated engineers to the projects, the projects nowadays bring work to the stable teams.
- Instead of pushing work onto the teams, the teams nowadays pull the work based on their capacity.
- Each team now had all skills needed to deliver tested functionality.
- Each team would work according the Scrum framework.

By doing this, the fact that some critical expertise was only at a few individuals became less an issue. The specialists were now part of a team and the conditions for knowledge sharing were set.

## **Moderate results**

The changes led to increased project transparency; schedule problems and technical dependencies surfaced earlier, which gave Sandra more options to react. The TU Processing engineers were quite happy with the new way of working, although they were still annoyed by the disturbances from the system testers which made that they could not manage to have a stable velocity, let alone an increasing one. Peter's management team was somewhat disappointed by the lack of results.

And last but not least, the various stakeholders were not as enthusiastic as was hoped for. Project managers missed focus on their milestones, and felt they had little influence on what was done when. Deliveries to the internal system testers were not more reliable than before. These were all signs of deeper problems waiting to be discovered.

Although Scrum had helped so far, there was much more to gain. The department did Scrum and faced the following challenges:

- Project manager success was still measured on compliance to plan instead of added value.
- Separate component teams optimized their own productivity causing longer overall cycle times.
- The responsibility of delivering an integrated product was still being handed over to separate integrators causing unnecessary coordination and costs.
- Stakeholders complained about the lack of project transparency as project progress was measured on individual components.

### *Teams need to take ownership and organize themselves.*

Like in the pre-Scrum era, most engineers conscientious did what was asked for. It appeared that only a few engineers had a drive for continuous improvement. This was not sufficient to have self-organizing teams with happier people and an increase in quality. The teams did not really live the agile values and have a significantly increased productivity.

Sandra decided to get outside help.

## **Emergent Innovation**

The department started working with Cesario from AgiliX and they decided to use the Emergent Innovation framework to adopt Scrum at large (2014). The Emergent Innovation framework is for product development companies seeking process innovation. It provides guidance for adopting agile development. Unlike prescriptive alternatives, Emergent Innovation engages employees and helps them discover the correct changes and innovations themselves.

### *Use Scrum to adopt Scrum.*

#### **Leadership change team**

It was decided that Peter's management team would lead the change. His management team became the evangelists, role models and coaches of the Scrum adoption. The team worked in monthly change iterations using Scrum to change the department.



Figure 1 : Initial innovation plan

One of the first activities was to create an initial innovation plan. The initial plan, as depicted in figure 1, included the objectives to reach, how to measure progress towards the objectives and what activities to start with in the short term.

## Change vision & shared values

People that are thoroughly informed about the business objectives, change vision and progress towards the objectives become engaged, aligned and therefore will more likely contribute to the change.

Therefore, Peter's team had a number of shared visioning sessions to craft the initial change vision. The initial change vision was eventually written down in a few crisp and clear lines. With this initial vision a broad discussion about values started. Each value would be up for discussion during a period of time so the whole department could discuss what the values means in terms of behavior.

The next step was to communicate reasons for the change.

## Communicate the change

In the first months, the change team spent much time on communication. Bi-weekly walk-in sessions were setup for people to just walk-in and discover about the change at hand. For people with a packed agenda, numerous invite only awareness-sessions were organized to become aware of the way they do their work today, what can be improved and how Scrum would benefit them and their company.

## Learning with Optimization Teams

Adopting large scale Scrum is a problem to which the solution cannot be known a priori as every organization is unique. The best approach is therefore to learn and discover your own path by continuously performing experiments using the PDCA cycle.

*The people that do the actual work are in the best position to do the experiments and learn.*

The department needed to learn how to apply the agile principles and values. It needed to learn new agile practices and discover how to do them in their specific context. People from the Scrum teams formed into Optimization Teams to discover what to change and how to change it across the department. Various Optimization Teams were formed on e.g. Scrum Mastership and agile testing. The Optimization Teams consist of anyone who wants to learn and share knowledge and define good practices for teams to use within the specific Thales context.

## **Optimizing the whole by Scaling Scrum**

In agile development you focus on creating value early and often. Although the component teams were working hard, their focus was on delivering their component and not on delivering a working feature. The teams can only deliver value when they deliver an integrated feature across all teams. Therefore, the next challenge was to move the focus from optimizing the individual component teams' performance to a focus on optimizing the performance of the whole department.

*Focus on optimizing for flow instead of optimizing for resource utilization.*

The most obvious step was to move from component teams to feature teams, but the management decided this was a too big change shortly after the move from project teams to component teams. Instead the choice was made to scale Scrum to the department level.

Scaling Scrum to the department means that all teams work from one product backlog. In four-week Sprints, all teams work together to achieve a common Sprint goal and deliver an integrated product. A product backlog item is broken down into smaller items for one or more of the Scrum Teams including their relationships and dependencies.

With this change in the system, the dynamics in the department changed as the focus turned from local optimization per component, to a whole product focus. The learning had started!

*One Scrum means one Product Owner.*

Peter took the role of the Product Owner of all the products that were under development in the department. He became much more involved in planning and ordering the work for his department which improved his steering capabilities.

## **Sprint Planning at scale**

One Sprint planning starts the Sprint. Running a Sprint planning for 70+ people is quite a challenge. At first all line management, project management, project architects and Scrum team's representatives showed up for Sprint Planning, including the stakeholders from outside the department. A lot of time was spent on discussing details that were interesting to only a few of the

people present. The meeting was not effective and the scaling of Scrum was beginning to be seen as an inefficient way of working.

It was then decided to better prepare for Sprint Planning. The following improvements were made:

1. All features that are candidate to be in the Department Sprint are properly refined before Sprint Planning. A Definition Of Ready was introduced.
2. The Product Owner works with line management, project management, project architects and Scrum teams' representatives to order the Product backlog before Sprint planning, so that the most valuable work is at the top.
3. The Product Owner prepares a Sprint Goal.



Figure 2 : Sprint Planning Part II

The Sprint Planning meeting at large was also split into two parts. First, the Scrum teams confirm the Sprint Goal and backlog features with the external stakeholders.

After that, the Product Owner and Scrum team representatives add the features to the department Sprint Backlog<sup>1</sup> until a Scrum team is fully loaded and other teams can complete no other feature. The planning is based on the average department velocity achieved in the last Sprints.

The Sprint planning at large consists of the following steps:

#### Sprint Planning Part I

1. Prioritize the features based on Cost-of-Delay and cost of development.

#### Sprint Planning Part II

1. Identify the capacity of the Scrum teams.

---

<sup>1</sup> The Sprint Backlog is called the department Sprint backlog because it defines the work the department has to do the coming Sprint.

2. Pick the most valuable feature and adjust the remaining capacity from the teams.
3. Repeat until the teams' capacity does not allow any new feature to be completely developed.
4. The Scrum teams identify dependencies between them and how to resolve them.
5. With the work distributed across the different teams and the dependencies identified each team individually plans their Sprint and creates a Sprint backlog.

## Daily Scrum at scale

Every day the various teams organize themselves around the department Sprint Backlog in a daily Scrum with all teams. (This daily Scrum is called the department daily Scrum) It is here where the teams plan how to resolve inter-team dependencies and impediments. Next to that, the individual Scrum teams have their own Sprint backlog and also have their own Daily Scrum.

*The teams manage their own dependencies at the daily department Scrum.*

Line management and project management also attend to be updated on impediments that they need to resolve. Having a daily Scrum at large with lots of people introduces the risk that the meeting will turn into a status meeting instead of a work session. Good facilitation prevented this from happening.



Figure 3 : Department Sprint Backlog

## Backlog Refinement at scale

At least two times per Sprint, project architects, integrators, domain architects, domain experts and Scrum team representatives come together to refine the top of the product backlog's features into stories for the individual Scrum teams.

After the features are broken up into stories, it is clear which team will do which story. Therefore, the individual teams can refine their stories independently of each other in their own refinement sessions so that the top of the product backlog is ready before Sprint planning. So there are refinement workshops for features at the department level and refinement workshops for stories at the individual team level.

## Sprint Retrospective at Scale

Every Scrum team has its own Sprint retrospective where they look at team improvements and team building. It is essential for teams to keep doing this with the people they work with on a daily basis.

*Management actively solves impediments outside of the teams' reach.*

Next to that there is also an overall retrospective to improve the performance of the department as a whole. Impediments that the teams cannot solve themselves or that have cross Scrum team impact are solved at the Department Retrospective. The Product Owner and Scrum teams' Scrum Masters participate in this retrospective.

## Sprint Review at scale

At the end of every department Sprint there is one big Sprint Review. The department Product Owner Peter presents to the external stakeholders the achievements of the last sprint. Typically also project managers and system testers attend, next to the TU Processing project architects and line management. The Sprint Review usually ends with the demonstration of one or a few realized features. Peter also presents the current department Product Backlog and uses the department Sprint velocity to relate this to the Project milestones. This gives the stakeholders the opportunity to take action if certain important milestones might not be met.

For practical reasons, the invitees are limited to the roles mentioned. However, the day after the Sprint Review, Peter updates in a 15 minutes session the whole department on the highlights of the Sprint Review session.

## Change to Servant Leadership

In the first weeks of the adoption management was discovering their new role. Management started with separate management sessions to discuss organizational impediments. They did see, discuss and delegate solving to others in the department. After a few sessions Peter and his managers realized that they should not be delegating but rather be solving the impediments themselves.

*Management understands the problems at the work floor through first hand observation.*

Management became part of the teams and joined the department Daily Scrum. They started solving impediments on a daily basis. The teams welcomed their commitment and transparency and as a result the teams had more respect and appreciation for the efforts of management.

## **Line managers become coaches.**

Before the Scrum adoption, each component team had a line manager responsible for the Scrum teams' success. Next to that line management was also responsible for coordination, people management, appraisals and budgeting. At the start, the line manager was also responsible to maintain the Scrum team's product backlog and decide on priorities as far as it did not effect the department Sprint backlog. This mix of responsibilities decreased ownership in the teams and inhibited self-organization.

*All people in the Scrum team are peers, there is no difference in authority.*

The improvements in the adoption of Scrum led line management to take a step back from the daily coordination and to pick up more of a team coach role. Their focus moved to ensuring the teams would never be blocked, and to coaching them on problem solving and improvement.

## **What are the results?**

As hoped with the introduction of agile, TU Processing achieved that the various stakeholders have clearer and more reliable steering information. Not only for the current Sprint but based on the feature point backlog estimation and the velocity, also for the future deliveries. This enables them to act on this.

Also the expected knowledge sharing was achieved: the number of critical individuals is reduced. In addition the unexpected disturbance for key individuals is significantly reduced which make that they can work more focused in the Scrum teams.

In addition, now the whole department focuses on the highest priority work, on the next delivery.

Last but not least, the quality of the work has increased. As a concrete example the integration of the integrated processing chain on the radar system could be achieved in a few days instead of several weeks. The total number of defects found after delivery has also decreased significantly.

## **Agile engineering practices at scale**

Agile development only works with the right engineering practices in place. Moving to agile development means delivering high quality software every Sprint. Software that can be changed fast and at low cost. The practices of for example pair programming, test driven development, automated testing, continuous integration and shared code ownership were a first step to produce working product across all teams every Sprint.

## **What are possible next steps?**

### **Team based incentives**

To increase the teams' focus on teamwork and cooperation, individual incentives are not enough. The next step is to introduce a team based incentive system where not only individual performance but also the importance of team performance is promoted.

## **Organize more around customer value**

Handovers from team to team, specialist bottlenecks and special roles like a feature integrator are sources of waste. At this moment they are still needed. A next step is to reduce these wastes by increasing the Scrum teams' ownership of end-to-end customer features. Coordination of work and handovers to separate teams should be minimized.

## **Maximize customer value**

The challenge is to increase the value created for the customer at reduced costs. This can be achieved by steadily increasing the current velocity, and to only deliver the features that the customer really wants, with the quality that is needed. The action here is to increase the collaboration with the internal customers.

## **References**

Ramoc C. 2014. EMERGENT: Lean-Agile adoption for an innovative workplace.

## **About the authors**

### **Cesario Ramos**

Cesario is an independent consultant that guides agile improvements throughout Europe. He strongly believes agile adoption can only be successful when the people themselves create the necessary changes and therefore are committed and feel accountable for them. He started working with agile teams in 2001 and nowadays works mainly at the organizational level as a professional agile coach and Scrum trainer from Scrum.org. Cesario is also the author of EMERGENT – Lean & Agile adoption for an innovative workplace.

You can contact Cesario at [cesario@agilix.nl](mailto:cesario@agilix.nl)

### **Sandra Ruijakkers**

Sandra graduated in Mathematics, and worked for over 10 years in various research projects in defense mission management and telecommunications. She gradually moved from doing technical work herself to project management and later on line management. She currently works as cluster manager at Thales, and she believes that the empowerment and stimulation of a team of capable people is the key to success