



Lessons learned: Using Scrum in non-technical teams

TOMASZ WYKOWSKI, ProCognita
JUSTYNA WYKOWSKA, ProCognita

Scrum is widely used in Software Development. However more and more non-technical teams are experimenting with Scrum as well. This article summarizes our experiences with using Scrum in non-technical teams in two companies in Poland.

1. INTRODUCTION

For over two decades Scrum has primarily been used by Software Development teams. Ensuring transparency gives them a number of benefits, such as the ability to focus on value, experiment, improve and quickly react to new knowledge. Given its widespread adoption, it comes as no surprise that non-IT groups have shown interest in Scrum. However their context, product and situation often differ from teams delivering a software product, therefore their path to agility might also differ. Over the last two years, we've been working with a couple of non-IT teams experimenting with Scrum. During that period we noticed a number of similarities, but also significant differences between software development and non-technical groups. In this report, we will attempt to summarize our experiences with Sales and Customer Support teams so that you'll be better prepared for transforming your organization.

We do not intend to explain Scrum ideas or elements here. Rather, we assume that you already know how Scrum works in IT and this report focuses more on differences and similarities, so you can use our stories when helping non-technical teams.

2. BACKGROUND

As Agile Coaches from ProCognita, we're supporting several companies in Poland. Since 2016 we have been experimenting with non-IT groups searching for improvements in their process by using Scrum.

The first of them, inFakt (<http://www.infakt.com/>), is developing an online accounting product. Established in 2008 it grew its business by automation and the simplification of accounting work so that Polish entrepreneurs could focus on their business. Of around 40 people 50% are responsible for software development, and the second half for other aspects of the business, including sales, marketing or customer support. After moving the whole IT department to Scrum in 2015 inFakt decided to start using Scrum in their Sales team to improve its effectiveness.

The second company, Codewise, (<https://codewise.com/>) is dynamically growing its international presence in internet marketing. Over a short period of time, it has grown to way above a hundred of people and found its previous working methods ineffective. Trying to maintain its non-corporate culture, Codewise provided teams with Agile Coaching but decided not to enforce any approach. In 2017 the Account Management team decided to give Scrum a try.

3. INFAKT, CHANGING THE WAY WE SELL PRODUCTS

In 2016 inFakt was struggling with sales efficiency. Four people were making a huge number of calls every day but the results were not satisfactory. At first, managers considered growing the team by hiring more people, however they were not sure where the problem lied, so they decided to check how efficient their process was. inFakt was already familiar with Scrum, as it had been used for over a year by IT teams, so they decided to give it a try in their Sales team as well.

3.1 How to add more work to an already busy team?

In contrast to IT teams, Sales have a lot of repeatable activities. In the case of our group, these were mostly phone calls to either new customers that might decide to switch to paid plans, or to existing customers whose subscription was to expire soon.

To improve the process we had to plan experiments and to run them we had to have some free capacity. We started by writing down all the work that was expected to happen over next two weeks. Then we asked the team members to select only those items that really must happen. Every single task was added to the plan, leaving us with no capacity for experiments. We decided to go ahead, and our first iteration focused on transparency and execution of already planned items. In the meantime, we prepared some experiments for the next Sprints.

3.2 Running A/B testing

For the next iteration, we had improved our Backlog and added some experiments to run. The biggest problem we were facing was lack of free capacity. Therefore the assumption we wanted to test was whether all the calls were necessary. We'd designed A/B tests for some groups, assuming that we had a large number of customers who would buy whether we called them or not, and also that there are people who are not interested, no matter how many calls they received.

Over the next few iterations, we proved our assumptions were right. In many cases, we were able to identify customers who do not need to be called to make their decisions. In other words, the chance of buying product was not impacted by the calls done by the team. Other calls could be substituted by automation on the product side, for example in the form of automated mailing or better tutorials. As the results of these experiments, over the next four months, we were able to free three out of four people. These members were able to focus on selling a completely new product to a new customer group.

3.3 How to show results when there's no product

As opposed to software development, the Sales team had no physical product to show and get a feedback on at the end of an iteration. Therefore presenting the sales results was more difficult than demoing working software. The first Reviews were rather boring, when we tried to show our guesses from beginning of Sprint and the final sales figures. There was no interaction with the audience. Moreover, the Sale Team has limited impact on customers decisions and the new product was difficult to sell, so the data shown was not stunning either. The team felt under pressure and uncomfortable going to Reviews. Therefore in one of the Retrospectives we discussed what a successful Review would look like and planned the next Sprint accordingly. Team members identified and interviewed key stakeholders on their needs and prepared the next Review based on the feedback received. Since then, Reviews have become more of a storytelling, rather than just showing dry metrics. The team began sharing important conversations (including playing some recordings), quoting customers, showing discovered patterns or common needs. Sales Review have become a great input for IT development planning, giving a better understanding of what the customers are looking for and what they're struggling with.

3.4 Taking over product responsibility

One of the most important changes Scrum prompted was the taking over of responsibility by the team. Before our experiments, team members usually worked individually with their lead, receiving assignments and executing them. Refinement sessions invited them to the conversation about what is important, what should be done and when, and what should not be done at all. They were involved in designing a conversation flow, where the goal was to understand customer needs rather than to sell the product. They evolved from a sales team into more of a customer support group.

Moreover, biweekly Reviews let the team focus on results, not task execution. One significant moment was when the team decided all by themselves to abandon an experiment in the middle of the Sprint because there were no visible results. Instead of continuing calls which were not leading to achieving the Sprint goal, team members decided to change their approach to contacting customers and running the conversation in another way.



3.5 Changing the salary model

One of the side effects of our experiments was a change to salaries. As in many other companies, people doing sales were rewarded based on number of successful calls. Reducing the number of calls the team members were doing took away the low hanging fruits for the bonuses they could get. In other words, team members felt punished for experimenting. As we wanted to attack the problem quickly, we raised it during one of the Retrospectives, letting the team know that managers are aware of it. Together, we discussed potential salary formulas and what could be impacting their wages. Based on our findings, managers modified team members' salaries. However, it took over three months from the moment their income dropped to the implementation of the new model. As one might imagine, this did not impact morale positively. On the other hand, in retrospect, people appreciated management's openness to discuss the difficult problem of salaries with the team members. Moreover, as the team lead noticed, people were way more motivated by the results and ability to make decisions by themselves than by sales provisions.

3.6 Transparency everywhere

Visualization is really common in Agile teams. However, what was interesting in the case of the Sales team (and other non-IT teams we've been working with) is their natural ability to quickly pick up physical tools. We've met many software development teams that had a hard time giving up using electronic ticketing tools which reduce communication and limit transparency. So far, we have not observed this tendency in any non-IT teams. Use of post-it notes, flipcharts and whiteboards seems to come more naturally to them. The Sales team ended up with their Sprint Backlog on the window, Product Backlog on the wall and all conversations taking place next to the whiteboard and flipchart.



3.7 Do we need a Sales team anyway?

One year of Scrum experiment in sales led us to many discoveries. First, we understood that most of our calls were not creating value and we could either automate this work or abandon it completely. Second, our customers were not looking for someone who wants to sell a product, but for advisors who will help them run their business using an inFakt product. As a result of this, after a year of experiments, we've slowly moved people from the Sales team to other groups, including Customer Support and decided to try Scrum there as well.

4. INFAKT, CHANGING THE WAY WE TALK WITH CUSTOMERS

The accounting business has a yearly cycle with the most significant workload in the first four months of the year. During that period the Customer Support group's main responsibility is to answer customer questions received through different media. However, from May the team has some free capacity that could be invested in other activities. In May 2017, looking at the findings we'd made with the Sales team, we decided to try Scrum in the Customer Support team. Together with the team, we planned the first Sprint. We wanted to focus on building a relationship with our customers. Identified work included articles for the blog, improving the knowledge base and the organization of a local Accounting conference.

4.1 Creation of an Education Team

Using Scrum in Customer Support gave the team ownership of planned and performed work by reducing managerial control and increasing transparency. This was noticed by both management and team members. People felt appreciated and as the team lead mentioned, "It's easier to share the results of your work when you've planned for them."

During the next eight months the Customer Support Team has successfully run seven events in different Polish cities. Over 80 blog posts and 6 videos recording were created as well. At the same time around 15 hundreds tickets were resolved by the team. As a result, we understand that our customers are looking for domain knowledge and appreciate the professional support from us. With this information, we decided to connect experts from different domains, including accounting and marketing. The new team was named the

“Education Team” and its goal was defined as the continuous accounting education of our customers. With the previous history of Scrum usage, the new Team decided to use Scrum as well and joined Customer Support bi-weekly Reviews.

At this moment, almost everyone in the company, except for a few people with internal support roles (HR, administrative, legal) are in Scrum Teams. The Customer Support Team is doing mostly repetitive tasks right now and plans to return to more diverse types of work from May.



5. CODEWISE – SUPPORTING FASTLY GROWING NUMBER OF CUSTOMERS

Codewise is strongly IT-oriented company operating in the online marketing business. The company came up through quite a well-tread path. As a small start-up, they’ve been able to conquer the market by quickly delivering products that address their customers’ needs, but over the time, as they have grown, agility has become a challenge. For a company with a couple of products and over a hundred employees, the delivery of new functionality or solving customer requests was taking weeks instead of hours. The organizational culture had also created a cult of technical perfection. The product was not seen as shippable until all details were implemented. Our proposal was to introduce an iterative approach to reducing the risk of over-engineering solutions. However some of software engineers had had bad experiences with pseudo-Scrum implementations from their previous jobs and opposed any changes to their current ways of working.

5.1 How to ensure SLA are met?

Interestingly, it was the Customer Support group who volunteered to run the experiment. The team was in difficult situation. There were two main problems they had been struggling with. First, the team was overloaded with requests, which was leading to burnout of employees and high rotation in the team. Several experienced team members had left the team. Second, with growing usage of the product, the number of customer request was increasing, and thus the service level agreements were difficult to meet. As a result, the team was trying to figure out an optimal group size which would be able to handle all the work.

5.2 How to visualize work if you have thousands of requests weekly?

The first step was focused on visualization. We noticed that there are 2 types of work items:

- Customer requests – hundreds of tickets logged daily. The team could neither control their arrival time nor amount at the moment. There were established SLAs for these items and therefore work should be started immediately on arrival.
- Internal requests included those generated by the software engineers, the marketing team, the management or the team itself. There were no SLA for these items, and the team could decide independently when to start working on them.

The team decided to use a physical board for internal requests and keep customers tickets in an electronic tool. They chose a weekly Sprint duration. The first iteration gave us some interesting insights. First, the items on the physical board were not moving at all due to dependencies, missing knowledge or just simply lack of time. Second, we noticed that the electronic tool was neither allowing us to answer the question of whether we had met the SLA, nor giving us the possibility of analyzing gathered data. As a result, the team decided to take two actions:

- reduce the number of items on the physical board by saying “NO” to some of them,
- identify and introduce a tool that would help us to analyze trends and patterns in customers’ requests.

Over the next weeks the team was pulling one or two internal request per Sprint, which allowed them to focus better and deliver those items faster. They also independently chose a new tool which meet their needs and implemented it. As it was easier to use, they were able to react faster to customer requests. Additionally, they were able to identify patterns and create a plan to address most common problems. This was a great sign that the team was taking responsibility for their own work, - a significant change from the previous way of working, where their tasks were assigned to them in the middle of the night.

5.3 How to work as a team, if you have “single person” tasks?

The team at this point consisted of 2 experienced people and 6 newcomers. Most work was done by individual team members, which resulted in lack of knowledge sharing, new team members being blocked by lack of information and duplication of work as the same problem was solved by several people. We wanted to improve the collaboration between team members, so the team started running Daily Scrums. In the beginning, meetings were pretty long as many blockers and issues were discussed. Over time, as the habit of collaboration outside the Daily Scrum evolved, they reduced the length of the meeting to under 15 minutes. Additionally a form of pair-programming, called “shadowing” was established – meaning that the team solved difficult issues in pairs. This gave the team members more experience and sped up the work of the whole team.



5.4 How to achieve flow in a disruptive environment?

Due to the nature of Customer Support work, it is impossible to predict or control the number of raised tickets. In addition constant context switching was impacting effectiveness of team members. Focusing on a single task was difficult when work was interrupted by new requests. To address that, the team introduced “no ticket day”. Each Sprint the team selected a couple of important items to be done. At Daily Scrum they decided who was going to work on this important improvement or task and that person was able to focus on this work, being protected from interruption by other team members who swarmed to solve upcoming customer requests.

5.5 How to automate the work of customer support?

Every Retrospective, the team analyzed data from the new tool, such as types of requests from customers, systems breakdowns, or new functionality delivered by developers. The team selected the most time consuming, most frequent or most hated type of requests and planned improvements to address them. Thanks to a “no ticket day” policy, some of the team’s work has been automated by creating tutorials, video guidelines, a self-study knowledge base and a repository of canned responses. This gave the team space for new experiments and freed team members from some of the repetitive work. From being full-time on answering customers the team achieved a balance of around 60% of time spent on customer requests and 40% for improvements.

5.6 Building a great Customer Support Team

A year since we began, the team has grown and split into more focused groups. All teams still use a weekly cadence, task board, and electronic tool, and the Daily Scrum is completed in less than 15 minutes. The practices of working in pairs and having a “no ticket day” policy have propagated to most of the other non-IT teams in the organization. Retrospectives and Planning have been combined into one meeting as our plans are mostly focused on improvements. The team decided that we are not only meeting our SLAs, but we are ready to extend our commitment to customers and now the team is going to provide support 24/7.

The team has prepared an introduction program not just for new members, but also for other teams and customers. New employees are introduced significantly faster than before the transformation.

Retention has improved significantly. During the last year no one has left the team and when team members got the opportunity to change departments, almost everyone decided to stay in Customer Support. In other departments this group is perceived as effective, delivering a good job, having most impact on product development and fun to work in. Other people are asking if they can join.

Finally, both management and the IT department started noticing the value of Scrum practices and any negative perception was very limited. IT teams working on the same product as the Customer Support decided to join the Scrum side. They began running two-week sprints and all teams, together with Customer Support run joint Reviews. The Customer Support team is invited by IT teams for Planning and is involved in the daily work of the software development group. The company is still growing and introducing new products. There is pull from other employees to move next products into similar synergy.

6. WHAT WE LEARNED

Working with non-technical teams has a lot of similarities with IT groups, but it shows also some important differences. During our work we observed several patterns:

- As in IT teams, forcing Scrum rarely works. Rather explaining Agile and Scrum values and principles; obtaining managerial support and providing help from experienced Agile Coaches gave team members time to learn and experiment.
- Safety is prerequisite for any change. However the job security in non-IT teams such as Customer Support or Sales is usually lower than in Software Development, therefore it is critical to protect the team, especially during the early stage of transformation.
- The product will be less tangible than in typical product development. Therefore, spending some time defining it and discussing how to present an increment in Sprint Review is very valuable. As are identifying your stakeholders, finding out what information they are looking for and analyzing, what you'd like to hear back from them. With this input you can better prepare the Backlog and plan your Sprint more efficiently. During Review we focused on storytelling, rather than just showing dry metrics. Instead of only discussing the work done, such as number of calls performed, we talked about results such as number of new sales. Share identified customers' needs and observed patterns.
- Scrum brings a lot of transparency, not just on what the team is doing, but also how difficult their work is. As a result, people can feel under pressure, but also appreciated by the rest of organization.
- With simple tasks, any estimation method more complicated than hours or T-shirt sizes would be over-processing.
- Using physical tools for visualization comes naturally to non-IT groups. With visualization comes a clear picture on what the team is working on, which leads to an evaluation of which part is bringing value.
- These teams are already fully occupied. Rather than adding new processes, identify work that does not bring value. Only after removing it can you find time for experiments and improvements. Try to automate as much of the repetitive work as possible or just stop doing tasks that bring minimal value.
- Involve team members in problem definition and searching for solution. Despite them being used to being told what to do, there is a lot of potential in these people that you'll discover as soon as you'll stop assigning work to them and start giving them challenges to solve.
- Moving the decisions to the team improves retention, ownership of work and pride in the job done.
- Non-technical Scrum teams incorporate new members faster.
- Review salary formulas and targets. Individual bonuses might be killing the team cooperation and willingness to abandon current practices and try something new.



7. ACKNOWLEDGEMENTS

We would like to thank all team members and managers that were working with us for:

- Being constantly attacked by IT vocabulary such as “refactoring”, “automation (build and test)”, “pair programming”, and using their creativity to translate these software patterns to the non-IT world.
- Having a lot of patience explaining to us what they are doing and answering all our “WHYS?”
- Open minds and lots of trust in this journey, we have been through together.

We especially would like to mention Wiktor Sarota, Marta Tymec, Malgorzata Szczepka and Ewa Wawrzon from inFakt for sharing their perspective on transformation; Tomasz Kaczanowski and Alek Fronczek from Codewise for inviting us to this experiment.

Finally, we'd like to thank Sallyann Freudenberg for her feedback on draft versions of the article and making sure we finish it on time.