# Parallel Juggling while Coaching Multiple Teams

IURII IALTANSKII, Accenture

Have you ever had the opportunity to work with multiple teams in parallel? How would you deal with too much coaching to manage at the same time? How would you balance out and take an individualized approach with each team while sticking to your values? This report describes what it´s like to work with multiple Agile teams simultaneously adapting to their needs and how “parallel juggling” worked out at a bank undergoing long-term agile transformation in Argentina.

## 1. INTRODUCTION

As a Scrum Master or Agile Coach, you probably know that working with your team can be quite challenging at times. Depending on the importance of the project, it might be exceedingly time consuming to attend to all the team’s needs and tackle all the issues. But what if you have 2 or more teams assigned? How should you treat each team? What would be the best approach to managing and planning your time so that all your teams are happy with the outcomes?

From October 2018 to December 2019 I was assigned as an Agile Coach/Scrum Master to work in one of the most important banks of Argentina (let’s call it ‘The Bank’). The name of the bank has been anonymized to protect their privacy. The Bank was in the middle of its Digital and Agile Transformation journey that started about 8 years ago. The idea of the top management at the Bank was to transform a traditional (in terms of organizational culture and hierarchical structure) bank into a vibrant modern organization which could provide a completely different experience of using banking products to its customers. To do so they decided to adopt and evolve an Agile way of working that resulted in the current ecosystem I will briefly describe.

### 1.1 Agile Ecosystem of the Bank

The state of Agile within the bank was a hybrid model with different frameworks and methodologies working together. It was complex in its nature due to the number of frameworks, methodologies and teams with different Agile maturity levels involved. Also, it required you to apply a lot of knowledge to get your teams to success as typically a large number of dependencies were involved with any particular team or project.
The teams could choose a particular Agile framework to do their job, whether Scrum, Kanban or a mix of different frameworks and practices. Three of the most popular Agile frameworks were Scrum, Kanban and LeSS. Some of the teams used Nexus or tried to adopt some of its parts.

The bank had about 50 Agile teams with more than 250 persons working in them. The portfolio of the Bank included about 50 different projects.

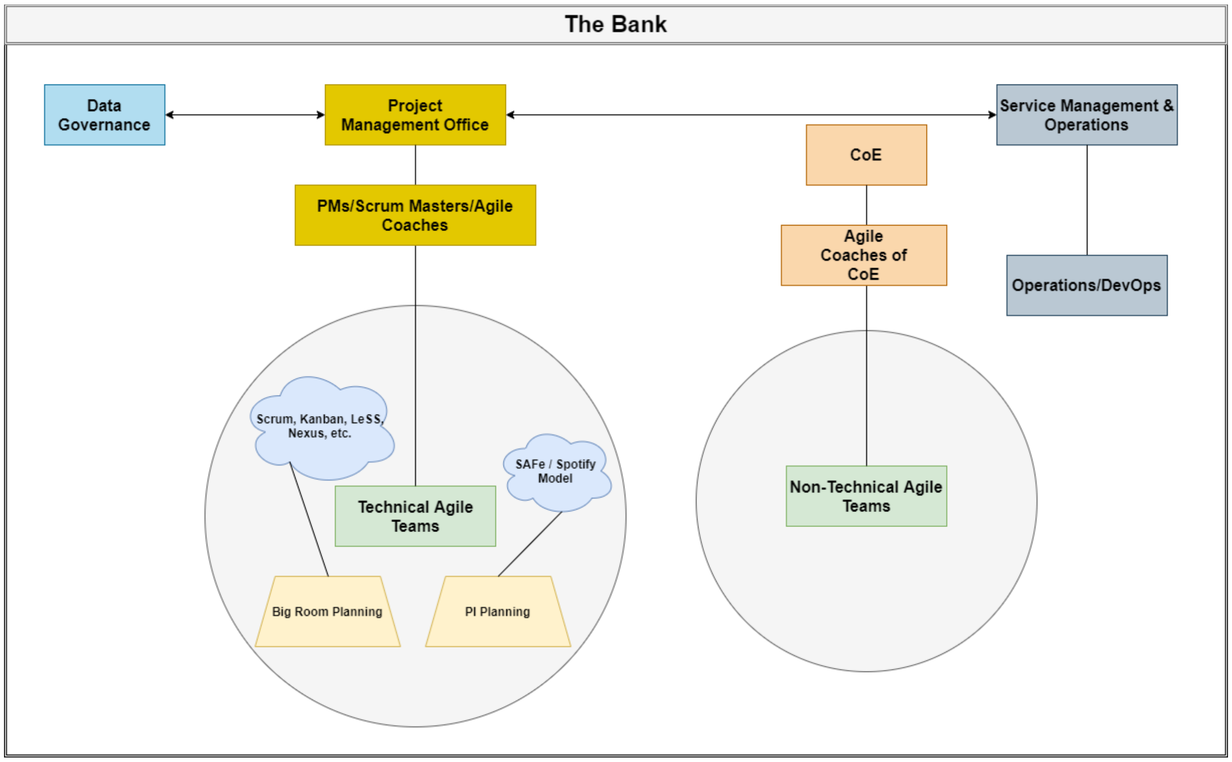
The Service Management and Operations department was managing all the Operation teams including Help Desk, Application Support and DevOps.

As it can be seen in Figure 1, the Bank had Agile teams working on different products like Core Banking, Web, Analytics applications, etc. Each team was dedicated to work on a particular product whether it was a fixed-term project or ongoing application development (like Core Banking applications).

The Project Management Office (PMO) of the Bank was responsible for Project/Program and Portfolio management of the Agile teams part providing the executives and CIO with status reports on each project to ensure synchronization between all the parts. The Center of Excellence’s (CoE) focus was on innovation as well as providing agile coaching for non-technical Agile teams. These teams were related to the business (Sales, Marketing, Graphical Design, etc.). The challenge the CoE had was to foster innovation within those teams helping to get their processes to the next level through the application of Agile methodologies and practices.

It’s worth mentioning that both the PMO and CoE had their own separate teams of Scrum Masters, Agile Coaches and Project Managers. Additionally, the Bank had a Data Governance department that was responsible for all data-related matters and collaborated with all the teams that worked on Data Analytics projects.

Author's address: Iurii Ialtanskii, email: iurii.ialtanskii@gmail.com


The Bank
Data Governance
Project Management Office
Service Management & Operations
CoE
PMs/Scrum Masters/Agile Coaches
Agile Coaches of CoE
Operations/DevOps
Scrum, Kanban, LeSS, Nexus, etc.
SAFe / Spotify Model
Technical Agile Teams
Non-Technical Agile Teams
Big Room Planning
PI Planning
Figure 1. Agile Ecosystem of the Bank

## 1.2 BI Team: New team, fixed-length project

Over a period of 14 months I worked with three teams—I'll call them BI, Core Banking, and Marketing Analytics. Each had a different niche in the way of working described in the previous section.

After a few weeks of initial training at the Bank, I was assigned to work with the BI and Core Banking teams.

The BI Team was a brand-new team formed to deliver a Data Analytics solution for the Back Office of the Bank. The project's time frame was defined to be from the end of October 2018 to April 2019. The team consisted of three Business Intelligence/Data Warehouse developers, a technical lead/Product Owner (Technical Product Owner), a Product Owner from the business side (Business Product Owner), and a Scrum Master/Agile Coach (that was me). Two people, including one of the developers and the Technical Product Owner, had previous Agile experience. For the rest it was their first time working in an Agile environment.

After the Product Owners and I had an initial conversation, we chose Scrum as our main framework and decided it was worth investing in a Sprint 0 to gather some information that was lacking before the team started development. I introduced the Product Owners to MoSCoW (this acronym stands for 4 different categories of initiatives/user stories: Must-haves, Should-haves, Could-haves, and Will-not-have at this time) prioritisation technique so it could be easier for them to define the initial priorities for what to work on. I also delivered training to the development team on the roles of the Product Owner and the Scrum Master and why we have them on the team as well as what is expected from them as members of the development team.

We had 2-week Sprints and I worked with both Product Owners to help them keep the User Stories well written. Usually, I scheduled a meeting to discuss with both Product Owners some stories where there were no clear Acceptance Criteria or it wasn't clear enough what is the goal the end user wants to achieve. I facilitated conversation between them to get those stories well refined. Additionally, we had one Refinement per Sprint scheduled with the whole team.

Also, I advised the developers to ask more questions to the Business Product Owner fostering collaboration between them. It was really important as, in spite of having experience working with QlikSense (that was the main development platform for Data Analytics), they needed to get more business context and that's where collaboration with the Business Product Owner was essential.

I spent 70% of my time with the BI Team, as it was a priority initiative within the Bank and the rest of my time helping the Core Banking Team.

During the first three months we made good progress delivering a product with high business value, getting excellent feedback from the stakeholders.

Once the routine in terms of ceremonies and practices became well established, this freed me up to focus more on the other team.

As a result of the collective effort of the team, the project was delivered on time and within budget. During the last month of the project (April 2019) I helped the Technical Product Owner sort out the requirements raised by the Business Product Owner to be included in the scope for the next project.

The main challenge I had with the BI Team was keeping the Business Product Owner motivated. He was a user of the application and had no technical or Agile experience. The main reason he didn't have the level of engagement needed was that he was taken away from his normal work to be the Product Owner. This role was an added responsibility for him rather than a principal activity. To help him in his role, I collaborated with the Technical Product Owner to schedule extra meetings with the Business Product Owner where we discussed the current requirements. I explained to both of them some points to keep in mind while writing User Stories and defining Acceptance Criteria. I also coached them on several good techniques and practices. This included explaining techniques such as DEEP and INVEST.

The Business Product Owner never ended up juggling these two jobs as quickly as I would have hoped. He wasn't fast in providing the answers we needed, being too busy with his main job. Luckily, those answers were predominantly about some low priority items, so that wasn't a big problem for the team.

As for coaching/skill building with the technical team, sometimes I could help by giving some practical advice on technical matters. But mostly, I preferred to stay out of the technical conversations. The belief I always had was that as a Scrum Master/Agile Coach you need to show just the right amount of technical knowledge to build trust and respect with your team. Doing more than that will do more harm than good.

**Lesson Learned**: *Don't trust your initial suggestions completely*.
Usually, I make suggestions based on past experience, but it's not always the best way for the current situation.

A good example of this was when I started working with the BI Team. My first thoughts were that the Business Product Owner would be better at learning and applying the MoSCoW technique. I supposed his daily tasks were more related to prioritisation. But it was the Technical Product Owner who could learn and apply the technique much faster and efficiently. So, I shifted to work more with the Business Product Owner to help him to learn this technique that was completely new to him.

This is what I learned: have your initial suggestions in your head, but don't rush into applying them until more details and information are revealed. Then, test your suggestions for improvement by trying and then readjusting your expectations. Don't become wedded to a certain way of doing things but be open to experimenting.

## 1.3 Core Banking Team: Old team, old issues

At the same time, I was working with the BI Team, a Core Banking Team was also assigned to me. Typically, each Scrum Master/Agile Coach in the Bank had at least two teams assigned to them at the same time.

The Core Banking Team consisted of three Front-End and three Back-End developers, a Technical Lead, and a Product Owner. They were working on a distributed Core Banking web application.

At our first meeting they seemed noncommittal. I got a feeling that the team was tired from dealing with some problems they couldn't resolve. It looked like an interesting challenge to deal with, so I decided to accept working with the team. In fact, the organizational culture of the bank supposed that any Scrum Master could accept or decline working with a particular team with no problem. The same principle applied to the team. So, the team could accept or reject a Scrum Master in case they felt he/she was not a good fit after the first few meetings.

Within the Bank, rotation of the Scrum Masters was somewhat natural due to a large number of the teams. Anyone who was tired working with the same team or just wanted to try a new challenge could ask to switch teams. Also, there were a number of the teams that worked without any Scrum Master or Coach assigned (no matter if it was a mature or rookie team) due to the relatively small number of the Scrum Masters available. It could be said that the Bank was constantly hiring new people to join the Scrum Masters team.

To get another perspective on the Core Banking Team's situation, I arranged to talk with their former Scrum Master. He told me the Core Banking Team was a mature, stable team without any rotation of team members in or out over the last few years. His perception was that there was a certain degree of reluctance on the part of the team to address potential areas for improvement. This was my perception as well. He also mentioned that the team's overall understanding was that they had room for improvement and could get their Agile practices even better then they currently were. The team also thought they could improve if they could apply new technologies in their development work which would increase their velocity. However, as there was a constant shifting of priorities on the stakeholders' side, the main focus for the team was on maintenance work instead of working on new functionalities. My opinion was that this lack of any significant new development of work, rather than any real need, had led the team to believe that some new technologies should be employed to improve their day-to-day work. Also, the team didn't like the recent decisions made by the Bank about the future of the application they were working on. The former Scrum Master thought it was better for the team to maintain the same pace without making any big changes.

It's worth mentioning that I did not feel that the team was in serious trouble. But it was my strong and sincere desire to help them to improve.

A detail I discovered once I started working with the team was that no Sprint Reviews were scheduled. When I asked the team about that, they told me that no one could remember when the last Sprint Review was held. They said, "no one needs it." After holding a few retrospectives using 4Ls technique (where each team member gives his/her feedback on the following key themes: Liked, Lacked, Learned, Longed for), I decided it would be beneficial to try using the Spotify Health Check model to delve deeper into the team processes and discover potential issues that might be addressed. The key takeaways from that retrospective were that:

- The team was disappointed by the fact that there was not much new development work available for them. Their main focus was to provide support and maintenance of the existing functionalities. As this was a direction chosen by the Bank there was some uncertainty in terms whether it would even be possible to change anything to improve this point.
- Consequently, it was impossible to start discovering and implementing new technologies. This undermined the creativity of the team.
- No Sprint Reviews were held as there was uncertainty on the stakeholders' side due to a constant shifting of priorities and responsible persons in the corresponding department.
- The level of satisfaction with the Scrum Master/Agile Coaching services was good, so I could eliminate any suspicion that the aforementioned points were related to the previous/current Scrum Master.

After the Product Owner and I had a conversation with the team, we decided to arrange an internal demo twice a month so each part of the team (Back-End + Front-End) could have an opportunity to see what's going on each side of the application and understand how things worked together.

The Product Owner also told me he would talk with the stakeholders to see if it would be possible to make a tweak in the current priorities, leaving some room for the development of new functionalities instead of working almost 100% on maintenance-related tasks. However, he warned me it would take time as we were close to the holiday season. So the team would follow the same direction in terms of working on maintenance-related tasks until January-February 2019. The team had some opportunities to work on new functionality before Christmas, which positively affected the overall team spirit.

In January 2019, the Technical Lead left to work on a different project. Also, in February two developers left. They were soon replaced. Over the next few months there was no clarity as to the longer-term direction. The Product Owner informed me that there was yet another shift coming on the stakeholder's side. The only thing we could do was wait until things were sorted out.

In April, the Product Owner left telling me he would be replaced with another Bank employee who previously had worked as Functional Analyst in a different product team. I couldn't say the team was glad to hear this news. As we started working with the new Product Owner, I delivered various training sessions to coach the Product Owner on Agile and Scrum roles and events, as she was new to Agile methodologies.

In July, two more members left. By August 2019, the rest of the team was reassigned to work on a different product within the bank. The maintenance work still went on, assigned to new team members, who had been reassigned to the team from other projects. And more new hires were added.

What I did with those new team members was deliver some introductory trainings on Agile/Scrum foundations and estimation techniques. As for the rest of the team, there was no need to introduce them to Agile and Scrum, so they just started maintenance and development work right away.

This was a common situation at the bank—reforming teams—similar to the situation with the BI Team. But in this case, it was a bit more intense in terms of coaching as most initial team members on the Core Banking Team had previous Agile experience. They had biases built in from years of experience. I couldn't just tell them to “try out a particular practice.” I had to provide more details on why the team would benefit from the particular practice.

One of the initial suggestions I got from the team was that they’d like the retrospective dynamics to be diverse without sticking to the same format for a long time. So I did, bearing in mind, however, that overly complicated techniques might be inefficient in terms of the outcomes. The team might be confused by an overly complicated retrospective ceremony and little to no improvement actions would be identified. So I tried to keep the techniques varied, but simple at the same time. In the end, the work with the team went smoothly for the rest of the time I worked in the Bank.

**Lesson Learned**: *Be prepared emotionally*.
More teams mean more people and more communications in your daily work routine. Sometimes these communications are fun and engaging, sometimes not. Be prepared for tough conversations on a daily basis. Here is where your Emotional Intelligence skills will be put to test. The approach I developed working with many different people (and personalities) at the same time, was to use my active listening skills. I don’t try to rush into a conversation until I’m sure I understand well enough the positions of my colleagues. The more I listen actively, the better the conversations went and the less chances any negative outcome would pop up.

I was able to apply my skills in conversations with the Core Banking Team during numerous retrospectives and was able to quickly identify their needs and worries. I found that some team members were shy to speak up during the retrospective. I made sure I asked them to talk first and, asking some additional questions, and found this made it easier for them to formulate their thoughts.

I also benefitted from participating in Agile Coaching sessions we held internally within the Scrum Masters team. It was helpful to bring up any issues I had and get advice. But also I could act as a Coach for others.

**Lesson Learned**: *Your initial plan won’t work out*.
Even if my initial plan seems perfect, it doesn’t mean it will work out perfectly. Especially when I’m in a position where change is constant.

As in case of the Core Banking Team, the organizational changes eventually led to 100% of the team members being replaced. Expect to have to go over the same ground in different new ways when things change. Be ready to navigate the sea of organizational changes to help your teams to weather every storm.

## 1.4 Marketing Analytics Team: Mature team, fresh wounds

After the project with BI Team was delivered, I was asked to take another team (I’ll call it Marketing Analytics). They were a mature team working on a Data Analytics product for the internal Marketing department.

Their current Agile Coach at the time asked me to assist with a retrospective. This let me observe how things were going as well as to get introduced to the team members and Product Owner. My first impression about the team was positive. However, I noticed that despite having a good grasp of knowledge about Agile methodologies and processes, the team wasn't happy with their retrospective’s dynamic. At the retrospective, no improvement actions were suggested. And from the conversation, it was clear that the team already had a pile of improvement actions pending with no progress being made on them either.

After the meeting I asked the Product Owner for a word in private to discuss the situation within the team. He told me that, indeed, the team had been working with Agile for a few years, but there were still some blind spots that they’d like to understand better. He also told me that since last year they’d been slow to improve their processes and no significant results had been achieved with their previous Agile Coach. This Agile Coach was leaving the company, so they wanted to have someone coach their team who could take a fresh look and bring up ideas to revive the team's spirit.

One of my first questions to the Product Owner was to ask how the team visualized their improvement backlog. It turned out they never had a way to visualize the aspects they wanted to work on. I found this surprising.

Another concern I had was that the team wasn't satisfied with their previous Agile Coach. They saw him more as a troublemaker rather than a servant leader. The thing I witnessed while assisting on some of the team ceremonies was that the retrospectives were so complicated that the team lost sight of their goal of identifying improvement possibilities in the process.

The team was frustrated that they got no valuable outcomes from their retrospectives.

In my opinion, this was due to complicated structure of their retrospectives and no tools available to gather the feedback from the team in a clear and concise manner.

So, the approach I used with the team from the first retrospective onward was to gather all the feedback to capture improvement actions. The easiest way to do that was to create an additional column in Trello (the main tool that the team used to manage their work) called "Improvement Actions" and track these improvement actions.

The last (but definitely not least important) area for improvement was that each Sprint the team was barely able to deliver around 50% of what was planned for during the Planning session.

I decided to tackle this problem gradually. Before making any changes, I wanted to make a thorough analysis of the current team's situation. A typical Trello board of a team would have three main columns (To Do, Doing, Done) and a History section arranged where the previous Sprint work done could be found.

The first step I took was to analyze the last five sprints to see if I could detect some patterns that might impede the team from improving their processes. I found out that the team consistently delivered around 50 to 60 story points each Sprint, while their originally planned Sprint Backlog contained 90 to 110 points. I gathered all the necessary data and prepared a presentation for the team. At that meeting I demonstrated clearly this negative trend and we discussed how it might be overcome.

The easiest approach would be just to grab 50 to 60 story points each Sprint while slightly increasing the amount of points over time. Team members had different opinions about what to do. Some were worried that this approach would endanger their velocity and the team could end up working even more slowly.

I explained that planning their work this way would allow them to change their negative outlook to a more positive one as they would see around 80% to 90-100% of their planned work completed by the end of the Sprint. And if they did increase their velocity, they might negotiate with the Product Owner some extra items to be done during the Sprint to add even more positivity. Some said that doing so would endanger the actual velocity and the team might even get slower. My response to this worry was that it would hardly affect the current velocity as the low numbers they had in the previous Sprints were due to the team committing to deliver a number of items that was above their current capacity. I suggested that instead of being upset with the actual results, they could try to start with fewer items and, if necessary, negotiate with the Product Owner to add more if they had time to complete them. Having fewer items in the Sprint Backlog would alleviate the emotional pressure within the team. After listening to my explanations, those who had some doubts agreed to try this approach. They were open to experimenting. However, we agreed that if things wouldn't improve in a few Sprints, we would review our approach and make adjustments based on what the team learned.

Next, I arranged a Starfish retrospective to gather more data from the team about their current practices. The team's feedback was that they never understood the overly complicated retrospectives conducted by their previous Agile Coach. Also, despite several years of experience working in an Agile environment they were still unsure about the estimation technique they used.

I noted was that there was a tendency for them to use "2" estimation points for the majority of backlog items. So, to get them to think rather than just say "2", I asked the team to not use "2" points while estimating for a while and try to decide whether an item was actually a "1" or "3". In spite of this limitation being introduced, some would still use "2" points while estimating. In such cases I asked them to justify the decision and fostered conversations to discuss and find out the most accurate estimation for the item in question. The team did this for several Sprints until they were OK putting "2" estimation points back to the scale.

I have found that once a team gets the estimation using "1", "2" and "3" points sorted out, they start using the whole estimation scale more accurately. And typically, teams have few issues estimating larger items.

The team had a short DevOps cycle implemented, but the process was a bit hectic. Anyone who wasn't too occupied with the development work could start working on user requests, but their response time to those request wasn't very fast as priority was always given to the development tasks. Also, the team wasn't happy responding to similar requests over and over without having any common stock answers. It was clear that some attention to knowledge management could help. I proposed to the team possible two approaches we might take:

- Talk to the Operations department to arrange a training and after that reassign all the repetitive requests that could have a simple workaround provided through a Knowledge Base article (Technical Note, etc.) to the Service Desk team.
- Make some readjustments to the current internal process, so the team would have a clear way to manage user requests.

The Product Owner told me it would take a significant amount of time talking to the Service Desk team, so the better approach would be to try the second option. So, we agreed to choose two people in the team to work with the user requests only while not taking on any development work. Those chosen were those that weren't too annoyed by working with user requests. The team and I decided there was no need to talk to the Service Desk department as we could improve the First Response Time indicator (all the service metrics could be found in the internal Service Desk application of the Bank) by adjusting the way we worked. After a few Sprints the Product Owner and I had a conversation and reviewed how the team was handling user requests. We found out the team still had 20 to 30 requests each week on their backlog with no response. I thought this might be due to a burnout of those team members who worked exclusively on the user requests. The Product Owner proposed to experiment and we arranged a rotation within the team to work on user requests. The idea was to schedule a rotation calendar and each week have two team members work on user requests only without taking any development work. Doing this would allow the team members who were temporarily the "core service requests engineers" to switch back to development work to prevent burnout.

After our discussion, the Product Owner and I talked to the team about the experiment we'd like to try and they agreed it was an interesting idea. As the result of the experiment, the team improved their metrics and decreased the amount of the requests pending first response to 5-10 requests per week.

The feedback from both stakeholders and the team was excellent; everyone was happy with the change.

But most importantly, the team changed their opinion about the Scrum Master/Agile Coach (me) and my role within the team. The team was happy they could double their velocity, improve internal processes and overall team spirit. It was clear to everyone that this was the result of the collaborative effort between the Scrum Master, the team, and the Product Owner.

I chose an approach that worked well with this team: whenever I delivered some negative news/facts/analysis results, I supplemented this news with a suggestion about how this could be solved/improved/mitigated. I was prepared to respond to any of their questions or concerns. This helped the team to embrace the suggested changes and built trust between us.

Usually, when I come up with a suggestion, I give the team freedom to give me inputs. However, there can be situations where the team doesn't have many inputs, either positive or negative. This can happen when the retrospective meetings are too complex to get the team straight to the point of identifying improvement actions. In this situation, I applied the same principles as I used with the Core Banking team: keep it simple and straight to the point. Only use complicated techniques when there's a real need. So, when I found with the Marketing Analytics team, the team was too tired from complex retrospectives arranged by the previous Agile Coach, I thought they'd like to have a simpler and more straightforward approach. And that's why, I guess, there were no objections to the approach I suggested.

I find that showing my openness to dialogue, real support, and sincere desire in helping them to get better in their Agile practices was essential to building trust. Also, I realized and was prepared to expect that there will always be a few that won't be convinced by my speeches, responses to their questions, or the suggestions I bring up. I can say that in the majority of the cases I had ready answers to their questions. However, for some cases like the one about using "2" estimation points, I had to explain in detail why it's important to distinguish between "1" and "3" estimation points to be more precise estimating in order to improve the velocity of the team. In case there was a conflict between the team members' opinions, it depended on the number of persons who didn't agree with the proposed action plan/improvement action. If those who didn't agree were outnumbered by those who did, I started asking questions to find out the reason why someone was not okay with the proposed action. Then I got all the team involved into a dialog and together we discussed the matter in question until a mutual agreement was achieved. In case the majority of the team wasn't okay with the proposed actions, I preferred scheduling another meeting, so the team could have some space to discuss everything in detail. That was important, because I don't think it's a good practice for a team to discuss some important issues during their Daily stand-ups. I see these discussions as a great distraction from getting on with daily updates, so my approach of scheduling separate meetings kept our stand-ups focused.

**Lesson Learned**: *Take your time investigating your new Agile team.*
I took my time observing the Marketing Analytics team and its internal processes. I talked to people and asked questions to clarify anything that was unclear to me.

Don't be afraid to do the same thing and you'll be good.

Experimenting, asking questions, and suggesting fresh new things to the team are approaches that both the team and the Coach benefit from. In fact, these approaches allowed me to grow as a Coach during my time at the Bank. While time management is important, switching contexts can be hard, too. If you are spread too thin, you might not have enough "free" time to have the necessary conversations to find out what is going on.

## 2. PARALLEL JUGGLING

It's uncommon for an Agile Coach or Scrum Master to work with multiple teams simultaneously. And when it happens, it requires not only lots of knowledge and practical experience to drive those teams towards excellent results, but also time management skills to organize your time efficiently. Let me share some lessons I learned doing my "parallel juggling" exercise while working at the bank.

### 2.1 Juggling Lesson 1: Plan your time well.

When I have one team to work with, I won't be too preoccupied about scheduling the events in my calendar. It would be just a Daily stand-up (that you set as a recurring event), a Planning, a Refinement, a Review and a Retrospective—all of these events occurring once per Sprint. After that I would have some additional meetings with the users, other teams, etc. My presence at these types of meetings would most likely be optional. And that's it, I'm good to go to use the rest of my time to plan my day as I like and find ways to help the team.

When I have multiple teams, I should be careful while planning all the events for all the teams to make sure they are not overlapping. Once, I had two Sprint Reviews scheduled at the same time and date, so I had to talk to the Product Owner from one of the teams to make sure he'd help to facilitate the meeting (it was an internal demo for the Core Banking Team) while I was at another Review where all the stakeholders were present (it was for the BI Team). All went well, fortunately. But after that I was very careful, meticulously reviewing my calendar to prevent this situation from happening again.

And, of course, the amount of my free time decreases dramatically. I didn't even think before how valuable my free time was. Now I do. Use it wisely. The key here is to plan and organize all the processes efficiently. Try to keep everything simple. Also, self-discipline is important, otherwise I would spend most of my free time procrastinating. First of all, I schedule all the essential meetings with my teams (ceremonies, etc.). Next, I try to organize the rest of my free time so that I can read something related to my job (it can be a book, an article about some Agile-related topic, a retrospective technique that is new to me, etc.). This helps me to find some new fresh things I could bring to my teams helping them to move forward. I don't forget about my personal projects. There's always a room (sometimes it can be a tiny one, but it's still there) for my personal goals, so if I manage all the things right, most likely I succeed in both professional and personal ways.

### 2.2 Juggling Lesson 2: Expect to work on different improvements with different teams

As I mentioned previously, I try to keep things simple. The same applies to the retrospective meetings. More simple retrospective dynamics allow the team to stay focused and generate meaningful conversations that would lead to identifying more improvement points. After the team has defined the points of improvement to work on, we decide on what would be the easiest points to start with and then adjust the pace as we go. A good approach would be to try to draw a parallel between the teams you have and see whether there are some actions/experiments/etc. that you could try with another team. Some of the actions that didn't work with one team could be successful with another.

After the initial pain points are solved, I found it becomes more complicated to find more actions where the team could improve. I suggest maintaining meaningful conversations with the team, listening to each voice within the team and changing the retrospective dynamics accordingly. The choice of what my next retrospective would look like is mine, but whether it would be efficient or not would depend on how attentive I am listening to my team.

So, those were the lessons I learned while working at the Bank. Currently, I'm working on a different assignment where there is just one team I'm working for. As the pandemic struck right after the project has started and we've had to switch to full remote context, I guess this could be a topic for my next experience report.

## 3. ACKNOWLEDGEMENTS

I thank my shepherd Rebecca Wirfs-Brock for her valuable comments and feedback during the Agile Experience Report Program shepherding process.