

# Adopting Scrum in a Management Team with a Just-In-Time Learning Approach

ERIK DE BOS SF Professionals

---

In this report, I describe an approach to instil Agile and Scrum knowledge based on a Just-In-Time strategy (CANEL, ROSEN & ANDERSON 2000).

By giving the team the bare minimum to get started with Scrum, and then helping them to figure out their journey driven by the problems they encountered and the questions they asked as they understood more and more of the framework, we found that the team picked up ideas more easily and with more eagerness than teams that received a full training in advance.

Fundamentally, this approach seems to lighten the cognitive load by creating a pull system for knowledge (DODGE ET AL. 2018). In effect, we approached learning by the same principles of Scrum: we prioritise the work (in this case, the learnings) and then focus on each item sequentially. This allows the team to take ownership of each new piece of knowledge, and gradually over the complete framework. The team becomes the driver of change.

Additionally, this approach reveals a structure for Scrum Masters to better decide when to help a team, and when to let them struggle, based on the risks involved. On the one side, the value of the struggle in providing an effective learning experience for the team. On the other side, the risk of a team struggling too long and ending up demotivated.

---

## 1. INTRODUCTION

In the summer of 2021, I was lucky enough to be asked to coach a team of managers to help them discover Agile. They were having trouble achieving their business goals and were curious about whether Agile would help them. The management team consisted of 12 franchise managers, each responsible for running their own shop, In other words, they were experienced experts in their field, with status and authority.

And they had no knowledge of Agile.

For me, this was a lucky opportunity because I had been working on an approach to help teams adopt Agile that relied on letting the team discover Agile gradually (DALMIJN 2022), driven by the problems they faced and with a flexible stance to the various standard approaches such as Scrum and Kanban.

To test this approach, The management team was a perfect fit. The fact that they had minimal knowledge of Agile would allow me to offer knowledge only when it became necessary. At the same time, their experience and expertise would ensure they would be critical about any new knowledge and not take everything at face value.

## 2. BACKGROUND

I have worked as a Scrum Master for the past 10 years. What drew me to becoming a Scrum Master was the idea of helping my teams to simply enjoy their work. We spend so much time at work, and it has such an impact on our lives, that making that time enjoyable seems to me an idea worth striving for. Watching a team in perfect flow is the perfect reward.

For me, Agile is a powerful way to improve work enjoyment, and I find Scrum an effective tool to help teams adopt an Agile way of working. However, I have concerns about the way Scrum adoption is organised. In general, teams are expected to adopt a predefined implementation of the Scrum framework in which everything has been thought out in detail. This can range from the form Scrum has in the mind of the Scrum Master all the way to the Agile Transition blueprint.

This to me seems overwhelming. There are simply too many concepts tied in a complex web of systemic relationships. Presenting these all at once is just too much to handle. The result is that teams usually start doing things because they are told to, not because they understand what they are doing. That is not an Agile way of working (ERINGA, BITTNER & BONNEMA 2022, DIKERT ET AL 2016).

## 3. ADOPTING SCRUM

Having accepted the challenge, I was given a whole day to meet the team and give them an introduction to Agile. First of all, we had a short discussion about the Agile Manifesto (FOWLER 2006). The goal here was to become aware of the kind of knowledge involved. Namely, that Agile is about a different way of thinking about the

problems we face, and not about some kind of ready-made solution that can be applied with minimal thought (DE BOS 2019). By having the team discuss the values themselves, it primes the team to accept that Agile is something completely different.

From there we jumped right into the Ball Point Game (GLOGER 2008), a fun, simple and engaging game that always works brilliantly to get people enthusiastic. People often get very competitive and fully immersed in the experience. That is the reason I like the Ball Point Game, because it illustrates the essentials of Agile in such a visceral way. A group playing the Ball Point Game is basically experiencing Agile in its purest form, without realising it.

### **The Ballpoint Game**

The game is simple. Take 100 pit balls and put them in some kind of container. Explain that the goal of the game is to pass, in 2 minutes, as many balls as possible through a system. The system is defined by the following rules:

1. one person takes a ball out of the container,
2. the ball must be touched by every member of the group playing the game,
3. balls must have air time,
4. balls may not be passed to someone to the left or to the right,
5. the ball must return to the same person who took the ball out of the container.

The group gets 2 minutes to discuss an approach. They then give an estimate of the number of balls they expect to pass through the system. This is noted down and the team try their approach for 2 minutes. After that, the group get 1 minute to reflect, after which the whole process repeats itself, starting back with the 2 minutes to improve on their approach.

With this group, the game followed the usual pattern. With such a clear goal, the team quickly figured out an approach and spent the first few iterations refining that and gradually improving their approach, encouraged by the short feedback loop of the game. Reaching a maximum, they started taking the thinking part more seriously and really getting creative until finally conceiving a completely different approach, one that helped them double their speed.

Within 6 iterations they had gone from 23 balls to 89 balls. These kinds of results always make a big impression on teams. It helps make the subjects we discuss in the debrief very clear. In fact, the debrief becomes no more than a moment to simply give a name to what they have experienced so vividly. The learning is in the experience so that later, concepts such as goals, flow, motivation, leadership, empiricism and self-organisation, will always bring the group back to what they experienced in the game. This also applies to the ideas of the Agile Manifesto. It is by experiencing them during the game that the values of the Agile Manifesto become 'real'.

After a break, we worked on visualising the workflow. My goal was to create transparency on how value flowed through their process, a first step towards a way of work. Interestingly, this exercise went like a breeze because they had already worked on it before. It did trigger the conclusion that their main challenge was sticking to their desired way of working, leading to a lively discussion about the need for discipline and focus. For me, this was significant because it supported the idea that it was not a lack of knowledge that was holding them back.

In the last activity of the day, we made a list of goals they had for that year, the things they were struggling to achieve. I showed them how to dot-vote the list. We then selected the top 3, and split up into 3 groups to break those down, guided by one question: "What tasks do we need to do to get this done?" In 30 minutes we created a list of various tasks needed to deliver these top three goals.

In effect, we created a backlog. It is surprising how often teams get bogged down in doubts and indecision, simply because they get swamped by everything they have to do. I could sense the relief in the room as they discovered how easy it was to get a grip on their work.

And with that, I felt we had the necessary elements to get started: a basic feeling for Agile, a workflow, and a backlog. That is, a new way of thinking, awareness of how to do their work, and a way to manage it.

But also, a lot of enthusiasm and confidence. In fact, by the end of that long day, I could see the eagerness in their eyes. It gave me the feeling that my hunch was paying off. Even though alarm bells kept going off in my head about the lack of theory, I felt I was effectively replacing *knowing how to do* by *wanting to do*.

### 3.1 STARTING WITH SCRUM

The next step was to choose a way of work. I organised a workshop in which I introduced the Stacey matrix (Figure 1.) and Scrum and Kanban. Again, I kept things very simple and practical, trying to find the essence of each concept. Kanban I boiled down to Little's Law ([LITTLE 1961](#)): the very simple but powerful idea that the less work you have in progress (limiting WIP), the faster you deliver the work (Cycle Time). Scrum to different feedback loops (the events) and an iterative approach (the sprint) to enhance learning ([ARGYRIS 1991](#)). And that was it.

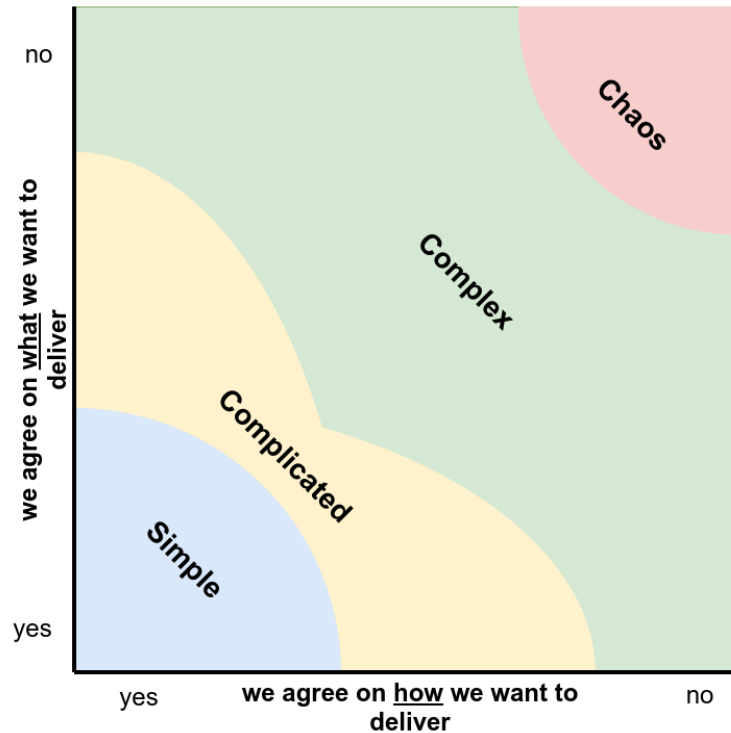


Figure 1. The Stacey Matrix, which describes how the level of agreement and the level of certainty define the nature of the work: simple, complicated, complex or chaotic ([SOURCE, STACEY 2002](#)).

After we discussed the Stacey matrix I was fully expecting them to choose Kanban, as we agreed that their work was not complex, but complicated. But to my surprise they chose Scrum. They saw it as an ideal way to give themselves a clear structure - the events - and to help them improve their discipline - the timeboxes.

I had prepared the rest of the meeting to get started with Kanban, and that was when I discovered that keeping everything so simple was also an advantage for me as their coach, as it was quite easy to switch midway through the session from Kanban to Scrum.

We discussed the Scrum accountabilities. The lady who had taken the lead in preparing and managing their long-term goals was an easy choice for the Product Owner, and another feisty lady who had shown a keen interest in Agile during the first session volunteered for Scrum Master.

We chose a sprint length and planned all the events in their calendars. And we chose Jira to support the process.

And that is where I lost them. Until then, Jira had always been my safety valve to help my teams release some steam and relax. You see, with developers, the Agile stuff is often the scary stuff and very much out of their comfort zone, but when we get to Jira there is always a sense of relief, as this is recognisable territory for developers: a digital tool with clear rules, screens and tabs and stuff.

Not so for a group of managers! As soon as we started looking at Jira, eyes started to unfocus, and big frowns started to appear. By the time I realised what was going on, our time was almost up. I scrambled to recap the Scrum plans we had made and hoped that that would be the positive note that stuck in their minds.

It was the moment I realised that my approach also very much applied to my own role. Just like with the team, it became clear that I should also avoid taking things for granted, and that it's essential to approach every new team with an open mind. Every team teaches you something new. If you listen and keep your eyes open.

### 3.2 OUR FIRST SPRINT

For the next session, I was all ready to do some serious damage control, but they had already fixed it themselves. The more digitally savvy people had helped the rest get started in Jira. What a relief!

So there was nothing more to do but get started with the first Sprint. We started right off with the Sprint Planning, where I just asked 3 simple questions.

#### *1. Based on your backlog, what is the goal for your first sprint?*

For me, this was the hard bit, because being managers, they didn't have anything remotely like "normal" Product Backlog Items (PBIs). We had a conversation about the qualities of a good PBIs and we eventually agreed that a clear deliverable was the key.

Again, a learning experience for me. We learn all these things about creating good PBIs: user stories (COHN 2004), INVEST (WAKE 2003), estimation (COHN 2006), and I was all ready to start preaching about those. But here, brainstorming with a group of open-minded people with a complete lack of knowledge about Agile, we came to a simple discovery: a clear deliverable can be an effective way to create a good PBI, one that naturally adheres to INVEST. A deliverable automatically makes a PBI **independent**. The concreteness of a deliverable makes the **value** easier to understand and makes **estimation** and **testing** easier. Of course, as soon as you can estimate, you can also decide if it is **small** enough and if need be, break it down further. And finally, the conversation about what the deliverable must be makes the PBI **negotiable**. Talk about keeping things simple!

With a clear idea of what we wanted to deliver, we then decided to work on multiple Sprint Goals in parallel, breaking the team up into sub-teams. This was another challenging moment for me as a Scrum Master. In Scrum, we want a single goal to promote focus and take advantage of the synergy of the whole team (RIPLEY & MILLER 2021), and the lack of a single goal is usually a symptom that can point at a variety of problems in the team (DALMIJN 2020). A single sprint goal becomes a tool to examine and solve these problems.

However, discussing this with the team we realised it was a challenge simply because of the size of the team, and the fact that their work was not complex enough to make it possible to work with so many people on just one thing. They showed a sharp awareness that this was a compromise, and they committed to be attentive to opportunities to collaborate closely between the sub-teams, so I decided to hold my peace.

#### *2. How are you going to do it? What needs to be done to achieve it?*

Here I was rewarded with a big surprise because these guys were surprisingly good at planning their work. Without any nudging from me, they defined milestones, planned their collaboration, and even discussed backup plans with a nonchalance that I found quite surprising. Until I realised that, being managers, that would of course be a given for them.

#### *3. All that? Are you sure you can get it all done?*

For me always a moment to start talking about capacity and estimation, but again, this was stuff managers are quite used to, so no challenges there. Except for the fact that they estimated in hours. I couldn't bring myself up to suggesting story points. I was a bit afraid of scaring them again (like with Jira), and to be honest, they did it in such a natural way that I felt it was worth exploring. I was curious about what I would learn.

And just like that, we were sprinting! It felt a bit like an anticlimax for me. It just all went so easily.

A few days later we had our first Daily Scrum. Yup, we were using Scrum for only a part of their work, the rest of their work consisting of the daily managing of their respective offices. They just found it too difficult to make enough room in their busy agendas to have a Daily Scrum every day. Also, they had limited time to work on their PBIs, so progress was slow. We agreed on two Daily's a week and seeing how that went. Again, I had to swallow a couple of times, but once more they showed such

awareness of the fact that this was a compromise, that I couldn't bring myself to make a big deal out of it.

The conversations during the Daily turned out to be really effective, which helped to mitigate the potential problem. They were clear and focused on their goals, and discussions were open and supportive. It felt great not having to go through the tiring old what-did-I-do-yesterday, what-will-I-do-today and what-impediments-do-I-have litany!

### 3.3 THE END OF THE SPRINT

Preparing for the Sprint Review, we discussed who their stakeholders were and they quite quickly realised that the people they managed were actually their most important stakeholders. That made them a bit nervous. They were concerned that a flawed review might create confusion and harm their leadership position in what was a rather traditional, hierarchical organisation.

After some discussion, they decided to do a few 'practice' reviews with their director to start with and wait until they had a Sprint Review where they had something clear to communicate to their people so that they could make a good impression on them.

Whatever the case, the impression they made on their director was great. She told them she was impressed by what they had managed to deliver, but also by their transparency regarding work they hadn't been able to deliver. And she was enthusiastic about their backlog and the choices they had made. Most important of all, at the end of the meeting she told the team that she felt she could finally stop worrying about them and let go. High praise indeed!

We ended with a retrospective where the group quickly focused on making their capacity more reliable, improving Sprint Planning and working more closely together. I could see they were enjoying themselves and were eager for more. As if this Scrum stuff was just like the ball point game, a simple game they felt confident they could play and succeed at.

I made some suggestions and they were graciously receptive to them, but I couldn't help but feel somewhat meddlesome. Almost as if with my suggestions I was spoiling their game. Stealing opportunities for them to figure it out themselves.

Sprint Planning came and I helped a bit with some discussions, but again with the suspicion I was getting in the way. A couple of times someone suggested lightening up on some aspect of the framework, removing a timebox for example, or skipping an event, but every time there was a teammate ready to remind them why it was important to stick to the framework. It was humbling. Where I had always found myself pushing my teams to stick to the framework, these guys were doing it themselves!

Later, the Scrum Master asked to meet me. We chatted about her role and I gave her some tips, suggested she do a PSMI. The real reason she asked to meet me turned out to be she wanted to do the next Retrospective. That's when I finally realised it was time for me to leave. They were telling me, perhaps unconsciously, that they didn't need me anymore. They wanted to continue without the safety net.

## 4. CONCLUSION

I left the company a few months later, so I don't know how things turned out for the team in detail. I do know that a year later they were still using Scrum, and seeing as this was a huge experiment for them, that is good enough for me.

Of course, doubts also haunted me after the experience. It had all gone so fast - 2 months! Was this just some strange exception? Or was it because they were managers, with a lot of necessary skills already there and just in need of a bit of direction?

I have tested this approach repeatedly since, with software teams, with similar positive results. It is an approach that seems to accelerate the human aspects of our work, things like safety, fun, self-confidence, trust. Increased team efficacy and improved value delivery follow reliably. I've even managed to help teams with a complex process and output-driven approach to rediscovering the art of simplicity, throwing so much baggage overboard.

These experiences confirm my suspicion that Scrum and Agile are not about knowledge, but about a state of mind. All that knowledge I drag around with me is just solutions to problems, best practices so to speak. Those don't make you Agile. A can-do, critical attitude and the courage to decide on your own

solutions: that is what Agile is really about. That attitude is enhanced by discovering the solutions yourself. The effort to discover those solutions creates ownership, they become unforgettable experiences. Milestones in the Agile journey. That is the Agile mindset (DENNING 2019).

Sure, it is great if someone's there to help out when it is a serious problem and things threaten to get out of hand. But ultimately, I feel that any time you get a solution without having to make the effort to find it yourself, it reduces the value of the solution. Or as in the case of the team in this story, it spoils the enjoyment. I've seen the same effect with my 5 and 8-year-old daughters, who will often get positively irritated with me if I try to show them a 'better' way to play a game.

At the same time, it becomes obvious why getting all possible solutions at once with no reference to a particular problem is not an effective approach. What you get then is a mass of knowledge without any context that you are suddenly forced to manage in your head. You become uncertain about how to use it. It leads to confusion and fear of getting it wrong.

I feel that the Just-in-time approach accelerates teams' ability to find and fix the problems they will inevitably encounter. It gives them confidence that they can solve anything that might pop up in their journey. It gives them ownership of 'their' Agile. This suggests to me that the approach is an effective way to *practice* the Agile mindset, as opposed to *learning* the Agile processes and tools.

For me personally, the approach made my work both more fun and more interesting. I spent less time preparing stuff and making sure I had all the answers, and instead focused much more on coaching: working together with the team to find solutions together. I felt much more a part of the team.

As I described in the text, there were various moments when I held myself back from telling the team *how it should be done*. This in some cases led to surprising insights. Such as for example, that Scrum is not only effective to manage complexity, but also to support discipline. Or the role of a deliverable in developing a good PBI. These insights into Agile and Scrum were a genuine pleasure. Just like with the team, by discovering these things ourselves, it made them more personal and stronger for me too.

Considering these results, the Just-in-Time approach reveals a useful structure for Scrum Masters, to help in finding the right balance between letting a team discover the problems and solutions for themselves or doing it for them.

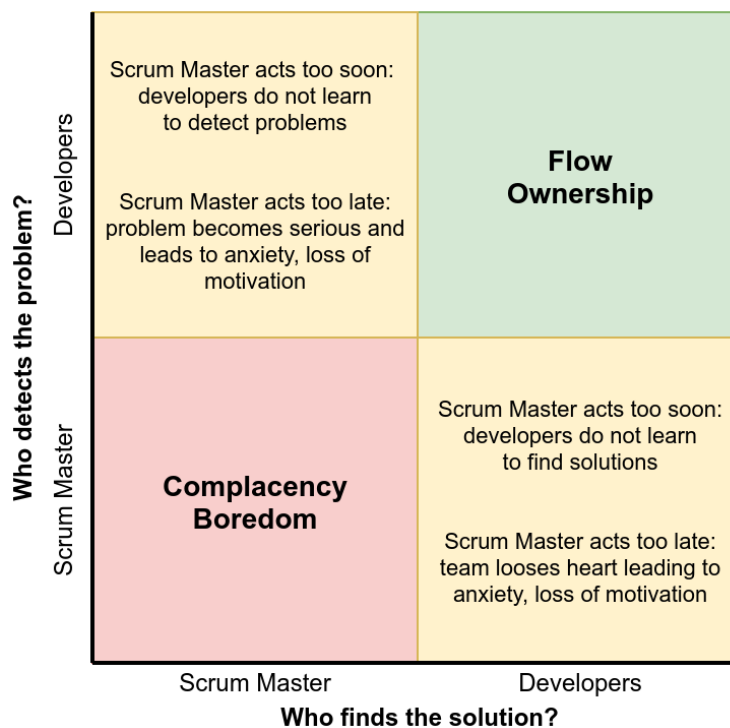


Figure 2. A matrix describing areas a Scrum Master should avoid (red and green) and the areas they should focus on (yellow), navigating a delicate balance between acting too soon or too late.

In Figure 2. I model this idea by using four quadrants, dividing spotting problems and discovering solutions among the Scrum Master and the Developers. In the bottom left quadrant (red), we see the Scrum Master both spotting the problem and delivering the solution. In the top right quadrant (green), the Developers do both. The yellow quadrants represent the areas where each accountability finds one or the other.

Here I see a parallel with Csikszentmihalyi's description of flow as a continuous balance between skills and challenges (Figure 3). The red quadrant corresponds to what he describes as boredom, while the green quadrant corresponds to the state that leads to flow.

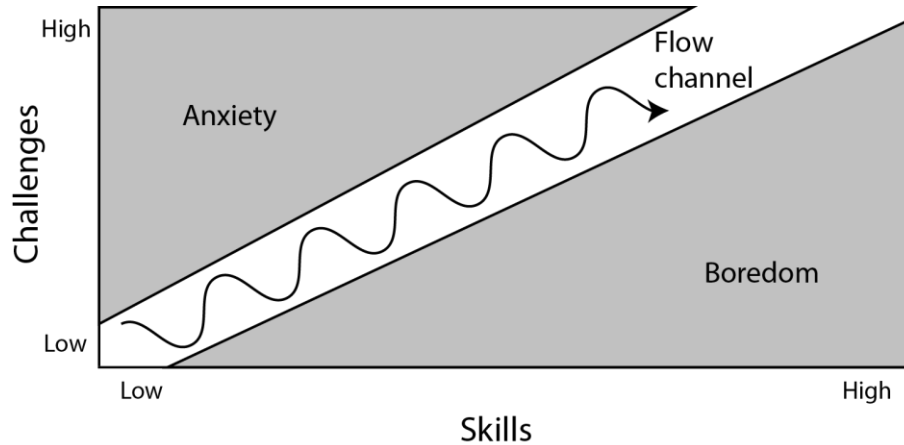


Figure 3. Flow as the delicate balance between the challenge and the skill of the practitioner (SOURCE, CSIKSZENTMIHALYI 2008).

A Just-in-Time approach effectively avoids both the red and green quadrants, leaving a Scrum Master to focus on the yellow quadrants, which are the critical areas where anxiety can occur and where the correct timing is crucial:

- Top left, the priority is to allow developers to learn to spot problems, but if this takes too long the problem can become serious and get out of hand.
- Bottom right, the priority is to allow developers to learn to solve problems, but if the team struggles for too long they may lose heart.

In both cases, waiting too long can lead to a loss of self-confidence, trust and motivation. The key to finding the right balance is a clear understanding of the risks the problem poses and how the team deals with anxiety.

So many things in Agile and Scrum are abstract and difficult to fathom, dealing with values and principles which are often hard to put into practice. I feel that a Just-in-Time approach together with this understanding of timing provides a concrete way for Scrum Masters to apply these abstract concepts effectively.

Take for example the Scrum Values. With a Just-in-Time approach, we show *courage* and *respect* by letting the team figure things out, and this allows them to build *courage* and *commitment*, helping them become more *open*, and *focus* on what really matters.

## 5. ACKNOWLEDGEMENTS

Many thanks to [Ziryan Salayi](#), for inspiring me to submit this piece. Also, to my friends at [Serious Scrum](#), whose reviews and feedback over the past years have made me into the writer I am today.

And of course to [Derk-Jan de Groot](#), for being my shepherd and guiding me into making this piece something I am proud of.

## REFERENCES

- Argyris C., 1991. "Teaching smart people how to learn". Harvard Business Review. 69 (3): 99-109.
- Canel C., Rosen D. & Anderson E.A., 2000. "Just-in-time is not just for manufacturing: a service perspective" [Industrial Management & Data Services](#) 100/2 50-60.
- Cohn M., 2006. "Agile Estimating and Planning". Pearson Education, Inc. 35-75. ISBN 0-13-147941-5.
- Cohn M., 2004. "User Stories Applied". Pearson Education, Inc. ISBN 0-321-20568-5.
- Csikszentmihalyi M., 2008. "Flow, The Psychology of Optimal Experience", Harper Perennial Modern Classics, ISBN 978-0-06-133920-2.
- Dalmijn M. 2020. "The inability to use Sprint Goals is symptomatic of serious Product Management problems", [Maarten's Newsletter](#).
- de Bos E. 2019. "What is Agile?", [Agile Thoughts](#).
- Dalmijn 2022. "Introducing Scrum Without Doing Scrum", [Maarten's Newsletter](#).
- Denning S. 2019. "Understanding The Agile Mindset", [Forbes Newsletters](#).
- Dikert K., Paasivaara M. & Lassenius, C., 2016. "Challenges and success factors for large-scale agile transformations. A systematic literature review". Journal of Systems and Software 119/16: 87-108.
- Dodge S. *et al.* 2018. "Pull for Knowledge Work", [MIT Sloan School of Management working paper 5380-18](#).
- Eringa R., Bittner K. & L. Bonnema, 2022. "The Professional Agile Leader", Pearson Addison-Wesley. ISBN 0-13-759151-9.
- Fowler M., 2006. "Writing the Agile Manifesto", [blog Martin Folwer website](#).
- Gloger B., 2008. Gloger is referenced as the creator of "The Ball Point Game" by amongst others the [plays-in-business blog](#).
- Little J. D. C., 1961. "A Proof for the Queuing Formula:  $L = \lambda W$ ". Operations Research. 9 (3): 383-387.
- Schwaber K. & Sutherland J., 2020. "The Scrum Guide 2020, The Definitive Guide to Scrum: The Rules of the Game" [scrumguides.org](#).
- Ripley R. & Miller T., 2021. "Should a Scrum Team Have Multiple Sprint Goals?". [Video blog on dZone](#).
- Stacey R.D. 2002. "Strategic management and organisational dynamics: the challenge of complexity". 3rd ed. Harlow: Prentice Hall.
- Wake. W., 2003. "INVEST in Good Stories, and SMART Tasks". [Posted on Agile123](#).