Small Scale Scrum

DR. LEIGH GRIFFIN, Red Hat
AGNIESZKA GANCARCZYK, Red Hat

How you scale a process like Scrum has received a lot of attention in recent years. Every time we think of scale we think of big numbers, more complexity, more coordination and more headaches to handle. We went and scaled Scrum, only in the opposite direction. Learn how we went about researching and innovating on the Scrum Framework to arrive at a variant of Scrum that we call Small Scale Scrum. Learn how we applied Scrum successfully to projects where we had 1-2 people available. The problems, the modifications and the journey that we undertook are now ready to be shared with the world.

1. INTRODUCTION

Scrum is increasingly becoming the leading framework of choice when a team embarks on the journey towards the Agile mindset. It brings with it a lot of benefits that teams, their stakeholders and their customers resonate with. Transparency in how things work, faster to market through releasable increments that deliver value early and often and simply having a closer look at how your product is evolving are some of the attractions towards choosing Scrum. Such is the popularity of Scrum, it has become the zeitgeist of our Agile times. Scrum has undergone a number of changes in recent years with modifications to the Scrum Guide changing the recommended team sizes. This has allowed flexibility for smaller teams to follow the Scrum Guide and achieve success. However, the Scrum Guide makes no suggestions as to how to run a Scrum team with less than the minimum recommended number, which currently suggests 3 people!.

Most research and innovations are focusing on how to run Scrum with multiple teams interconnected and coordinating as one. This is scaling in the classic sense, but how do you ensure the same success and quality in teams that need to scale down below the minimum? How do you deliver the value that the customer wants, while ensuring that you follow a process that they wish to participate in? How do you scale scrum, down? In Red Hat, we have a number of teams that deliberately work in small teams. Our Professional Services or Consulting teams work with partners and customers to deliver value through small, targeted teams. These are short term engagements, with minimal number of people working on them due to the budget being fixed. These projects need to deliver the customers goals, they need to uphold the quality and reputation of Red Hat and they need to start quickly and deliver value immediately. They also need to adhere to the guidelines and wishes of the Customer, which includes alignment with their process framework of choice. Our style of work was a prime target for innovation and became the main focal point for figuring out how to apply Scrum to small teams.

We also had other opportunities to innovate with smaller teams. Our Open Source contributions are delivered through 1-2 person initiatives on services and mini products in their own right. Our internship program brings us students who are required to complete a capstone project as a lone developer. What has become the norm for us is customers, our community and our partner universities demanding that we use Scrum. The perceived benefits from their perspective as our customers are enough to make this demand when engaging with us. This is particularly true for our paid for consulting work which forms the bulk of our experience. This insistence on using a framework that does not provide guidance for how to operate in such small numbers both surprised us and challenged us. It motivated us to research how Scrum could work in such limited scenarios and the result of this has been a 2-year journey to develop, iterate on and promote a way of working that suits small teams. This is our experience.

Leigh Griffin, Red Hat Waterford, Communications House, Cork Road, Waterford, Ireland. lgriffin@redhat.com
Agnieszka Gancarczyk , Red Hat Waterford, Communications House, Cork Road, Waterford, Ireland. agancarc@redhat.com

2. BACKGROUND

I will never forget the first experience of working in a Scrum team trying to adhere to the Scrum Guide. I was a fresh graduate developer working in a research center and our team was about to embark on a journey in Agility with Scrum at the heart of it. I, Leigh, couldn't believe the freedom that this way of working was bringing. I was being heard. I felt empowered and I started questioning why we were deviating from the Scrum Guide and how we could improve as a team. That experience was repeated for several years in different teams and companies and really culminated in me transitioning into a ScrumMaster. I felt I was a natural coach and my PhD was centered around how groups of people and services formed, with an emphasis on the social aspects of team formation. That insight allowed me to grow my career as an Agile enthusiast and over time I fulfilled all roles within the Scrum Guide. My PhD was also a turning point as it was a solo body of work that I simply had to deliver with no real support. I loosely following a Scrum based approach as I knew it would help me focus on Value and keep me honest in retrospectively analysing my papers and thought process with my supervisor. That was the first time that I delivered a Scrum enabled project as a single person.

My work in companies brought me into contact with student interns who from a retention perspective we would help support in their final year capstone project. I encouraged all students to follow some form of process to keep them sane and ultimately deliver their project on time and with high marks. Loosely following Scrum and the visuals of Kanban really helped me see the potential for applying a full blown team orientated framework on a single person project. It really piqued my interest but my research career ended as I moved towards pastures new of working in industry on more tangible real world projects with real paying customers. The long and winding road brought me to Red Hat in 2015 where as a People Manager with a specialism in Process Improvement, I have found my career best role. I get to manage a team that follows both Scrum and Kanban to deliver Products to our customers and indeed our Community. I get to work with people on their problems and help them become better versions of themselves and help the team to succeed. And I get paid for it, can you believe that?

This role saw me embark on several Agile transformations across teams and products. It also brought me back to working with college students as Red Hat places a huge value on connecting with local universities and helping students experience our way of working. This sees us engage in multi month work placements and donate our time to mentor final year projects. Seeing the stress and pressure that the students came under and having already had some mild success in this area of delivering a prolonged body of work in a solo mode, I experimented with modifications to Scrum to enable a student to plan, prioritise and deliver value through their final year project. The 6 students who followed this over 3 years received the highest marks in their class, with the reviewers commenting on the professionalism of the project's planning and execution. I had worked indirectly with Agnieszka in our office and over coffee one day I enquired about how her Masters was going. She was struggling for a topic for her thesis and we had connected about Agile initiatives several times in the past. My light bulb moment came and I pitched the idea of formally researching scaling scrum down. This led me, Leigh, to mentor Agnieszka in her Masters thesis.

I came from a very different background to Leigh and took a winding road into IT as a profession. I graduated with a degree in Law and embarked on legal professional career but never quite had the passion for it. I was always intrigued by software and problem solving and spent most of my secondary school days in the early 2000s programming. Unfortunately, that early experience didn't convince me to pursue it as a career and also the risk of moving into software development career versus a stable well paying job in Law was something that held me back. My husband had a similar thought process and when he realised that he had a real potential in computer programming career, he left his job to go back and take on a 4 year computer degree. Through his college experience, I was amazed how much software development changed from the early 2000s and that gave me the encouragement to go ahead and try and realise my once unwanted career and transition into software development.

Thankfully with an existing qualification and some prior knowledge of programming, I just had to complete a 1 year Higher Diploma in Computer Science. That allowed me to start in a software company specialising in R&D software projects which eased my entry into the IT industry and fundamentally prepared me for my new career. As a new person to this industry, but with great ambition to succeed and equally the fear and self doubt at times you would expect, I discovered that this team had a really open way of communicating and planning work. I didn't feel like I was inexperienced and my thoughts and opinions mattered. This Scrum process combined with my prior customer focused mindset was making my life easier and when I moved on in my career and joined Red Hat as a Consultant, I discovered a new way of working. As a paid for Consultant, we are billable by the hour or day. Customers want to maximise their value and their budget so the projects I worked on were now staffed with 1 to 2 people and a part time project manager on oversight. This gave me huge experience with various roles in a team as day to day you need to wear so many hats. My specialism was as a Software Engineer writing frontend code then switch to backend work and as a Quality Engineer every few weeks before final releases, then due to demand I briefly moved into more of a Quality Engineer role but day to day I could be writing frontend code and then switch to some backend work and finally into quality role.

Roles switching is very common and dictated by project needs, giving associates the maximum experience of delivering complete and high quality enterprise software products. Customers employing an outside Consultancy house ultimately have to own that product or functionality that you create. That means contractually we look at minimising the friction between our style of working and that of the Customer. We align on tools, on programming languages and on other technical elements. In recent years we are starting to see an alignment on methodologies. Our Customers often have appointed Product Owners and their teams are running as a Scrum team. As such, they want us to run in a Scrum manner. They love the pause and inspect opportunities as in a finite time project (typically under a month), the chance to inspect and adapt is crucial. Red Hat has worked on a number of improvements in methodology and process and our Open Innovation Labs has helped to make a much more fluid and Agile experience for our customers which helps to train teams in order to achieve the shared goals. Given my relatively accelerated entry to this industry I decided to go back to college part time and address some of the gaps that a 1 year transition course could never cover. My MSc gave me a lot of technical skills and coupled with my day to day I really gained the strength and confidence I needed. Now I had to do a capstone thesis, a 6 month or so research project. I really didn't want to go deep on another language or tool or tech in general as I was at saturation point! A chance coffee with Leigh and I find myself researching Scrum. I thought I knew a lot about Scrum but now I had to figure out how do we adapt it to work with these small teams. I had a ready-made test group in my colleagues and an enthusiastic mentor. What more could I ask for!

2.1 The formalization of Small Scale Scrum

In early 2018 we set about formalising Small Scale Scrum as Agnieszka began her research on her thesis area. We wanted to frame our research around a theme, that theme was Quality. Quality is the hallmark of a good project, it is what companies like Red Hat build their reputation on and sustainability of projects is non-negotiable goal. Agnieszka focused on this from the professional consultancy projects. We looked at the Agile Manifesto and the Scrum Guide and took a pass through it from the perspective of what modifications would be needed to represent the concerns of teams of 1-2 people working on an initiative. Working for a company like Red Hat that has very strong values and principles has really stood to us while designing and thinking our way through this topic. If you are ever trying to make a culture of innovation and Agility work in your organisation, consider what your values and principles are. They can make or break and initiative in our experience of having worked in companies that did not have such a strong guiding hand. Our way of working in Red Hat is both refreshing and novel and we would highly recommend a values-driven approach to how you, our readers, run your team.

Small Scale Agile Manifesto

The manifesto obviously does not recommend team sizes or implementation details and is very much at a philosophical level. Therefore we did not make any modifications and we wished to keep the spirit of the Manifesto intact. However we wished to provide some additional values to the to help emphasise the need to be more aware of the possible pain points and challenges that we had identified. The following are the values that we arrived at but we styled them in the phrasing of the Manifesto for consistency:

Wide Communication over narrow. We found that teams often use information radiators and have a heavy narrow focus within the team. That behaviour is harmful in small teams so defaulting to wider stakeholders to promote transparency is our recommendation here. We were very surprised to note how few communications come from a normal sized Scrum team on a day-to-day basis. We observed communication peaks at sprint boundaries but the teams were very insular outside of that. Our thoughts here were early and often communication to get out in front of any blockers or misinterpretations that might occur.

Team feature delivery over individual responsibility. In big teams, despite work being on a board and in theory anyone able to take work and progress it, specialists attract to their work area consistently. Leigh in particular observes this daily in his larger teams and mini silos of responsibility and singular points of failure start to emerge unless dealt with by a good ScrumMaster and an even better self-organising team. By promoting a team mentality of overall feature delivery, Agnieszka discovered on her previous projects that the very best Consultants would happily step out of their comfort zones and put the project needs first.

Quality delivery over speed of development. As quality was our main challenge we wanted to enshrine it in our manifesto. Rapid development is often at odds with quality delivery but finding that balance of best practices, good tooling and expectation setting was going to be the key for how we envisioned this working.

Multiple project responsibilities over fixed assignment. Agnieszka in particular wanted to reinforce that you can't have a singular role in a project where the full software development life cycle is to be experienced. As students, Leigh wanted to emphasise that the success of the project is all about switching mindset into another role and fulfilling that.

Accelerating innovation over marginal request driven thinking. Our jobs as Engineers is to help interpret what a customer wants and help them arrive at what they need. The temptation is to deliver exactly what they requested and follow that strictly. Innovation is what customers pay companies like Red Hat for and having the seeds of innovation planted within the teams thinking was going to be critical to the overall success and sustainability of the project completed.

Customer growth over customer engagement. A successful engagement is obviously important but we wanted our customers to grow and learn on this journey as well, given they are a key stakeholder. Growth or creation of business is what our customers highest priority is and it should therefore be the teams highest.

Small Scale Scrum Principles

We created four guiding principles based on experiences in small scale consulting projects to complement our Manifesto:

Value-Based Communication - With finite time, all communication should be valuable.

Quality-First Development - This principle focuses on taking a quality approach to software development each Sprint.

Delivery Ownership - This principle is about taking initiative in driving software delivery by a Development Team on a Sprint-to-Sprint basis.

Iterative Sign Off - Identifying gaps in requirements through an iterative sign off approach is something we found very important.

Small Scale Scrum Framework

Now that we had a guiding manifesto, principles and guidelines, we set about designing modifications to the Scrum Framework to arrive at our version of the Small Scale Framework presented in Figure 1.

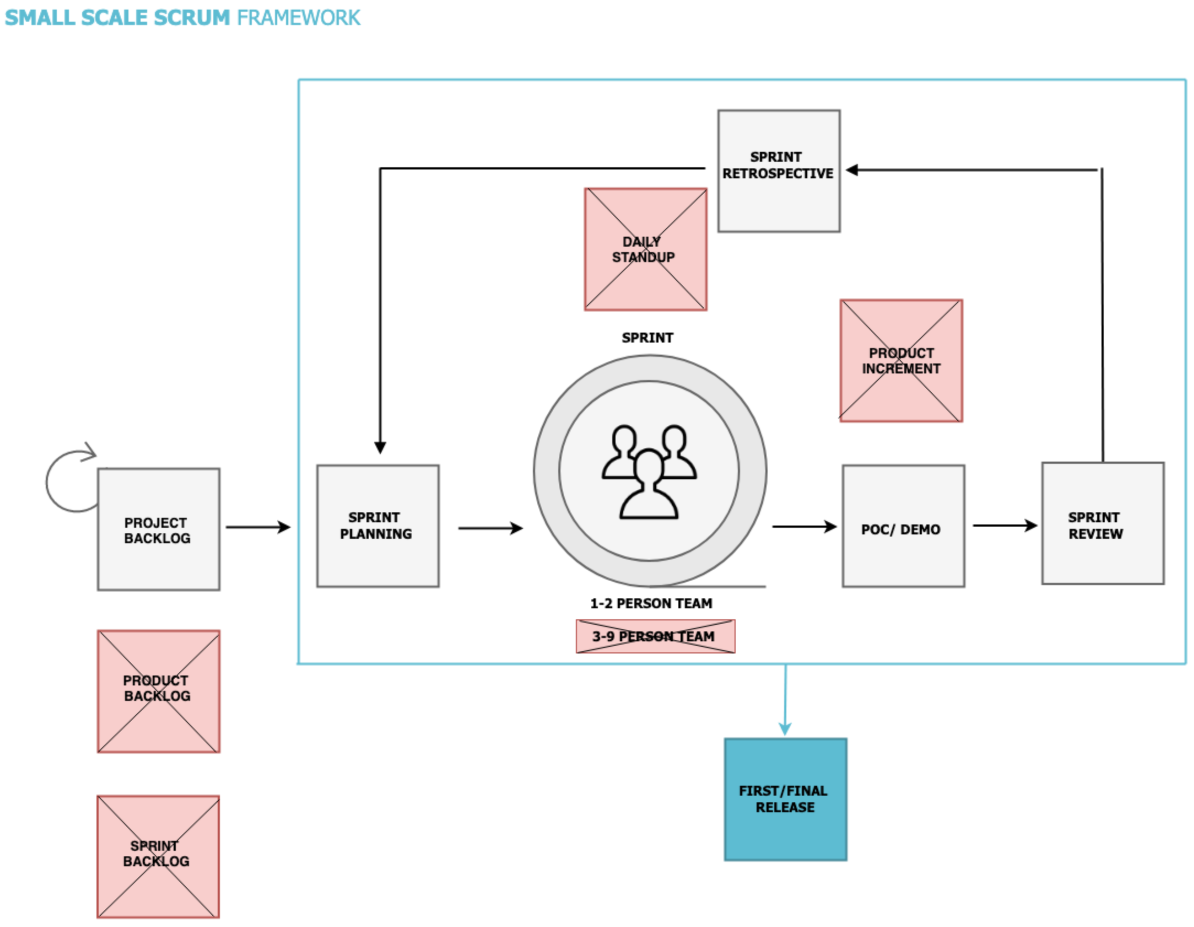

Figure 1 Small Scale Scrum Framework.

Let's focus on the changes that we had to bring in. First off, there is no Sprint Backlog and Project Backlog. In a small scale project, the duration of the project is finite, so we amalgamated those concepts into one singular Project Backlog. It is a list of everything that is known to be required in the project and is interspersed with User Stories and the technical tasks that are required to support and realise them. For us, this was hugely important for our consultancy projects as it highlighted to the customer how much work is actually involved in realising a solution. Often with our user stories we want to stay at the What and Why level of discussions but the How is really important here to make informed decisions in a time sensitive manner. The team really appreciated having their viewpoint shared so widely and knowing that the entire backlog was the known scope gave them a great sense of progression as we burned down towards our goals.

The sprint here is something that we recommended with an upper bound of 2 weeks but may be the order of days instead of a minimum of a week. The recommendation of no less than 1 week and no more than 30 days is not really workable in a small project where the timeline may actually be 30 days or less. The fluidity is needed by the team and the lightweight nature of the ceremonies around it allows for this shorter turnaround.

An interesting addition was that of the Proof of Concept (PoC) / Demo. Now you may be thinking that is the Sprint Review right? Sprint Reviews are where our stakeholders come in and have the accept/reject goals of the Sprint or more accurately the increment that is created. As the team may have very short and variable sprint durations to tackle key pieces of the project backlog, we wanted to have some flexibility here to have an internal PoC or Demo. This is very much a rough prototype phase to ensure that our direction is sane and that the team has much more flexibility in their creativity and can take chances that they ordinarily would not want to try with the more formal demo where a release or drop of the functionality is expected. As such, this additional step is important for the confidence of the team and enables us to innovate in a safe environment.

2.2 Informing our Approach

In parallel to our creation of the values and principles, we decided to take a broader look at Agile and Scrum to allow our findings here feed into our inspect:adapt loop. The actions taken included conducting interviews with Agile Practitioners running Scrum and Agile variants in large projects within Red Hat and running a "Small Scale Scrum vs Regular Scrum" survey in order to understand how people felt about using Scrum in their teams of varied sizes. We wanted to extract Scrum in its large context to see what could work for small teams in short term engagements. We were interested in what worked, what didn't work, what could we change/improve. We then fed survey and interviews results directly into Adaptation / Test of selected Scrum Ceremonies and Principles step of the framework's formulation process.

Some responses within the survey came as a surprise to us. A big one, which we have seen first hand in our own projects, is that teams do not adopt a Definition of Ready (DoR). This resulted in work commencing that lacked all of the information which ended up being gathered in flight. That presented obvious challenges which from our own personal projects that we analysed resulted in missed requirements and even bugs. It was refreshing to see that more mature teams had not adopted this and we dug a little bit deeper. Scrum teams that had not adopted it were generally not on a tight timeframe. It was fine for them to start work and use the sprint increment to try and sharpen up on the acceptance criteria. Their Product Owner was not a traditional Product Owner role. On going deeper again we found that a lot of Scrum teams simply don't have a full time dedicated Product Owner. One of the 3 core roles was simply not being fulfilled so the teams were following the framework but lacking the oversight, direction and value setting that a Product Owner brings. Pulling that simple thread of the DoR led us to realise that teams that self identify as being a Scrum team are not adhering to the Scrum Guide definitions. That gave our Small Scale Scrum adaptation a real boost as the modifications and tweaks we had to make would never be viewed as pure Scrum. Yet our version of Scrum seemed to have a lot more stability, value and quality than the traditional approach taken by the developers, Quality Analysts, ScrumMasters, Product Owners and management teams that formed part of our survey of Agile practitioners.

2.3 Our capstone project

After a lot of research, data gathering and trialing aspects of the approach through a number of projects, we had the version of Small Scale Scrum outlined above. We wanted to tackle a project in a capstone kind of manner to really get a sense of the benefits that Small Scale Scrum could bring. We led a final year college project from September 2018 through to April 2019 with a student called Ciaran Roche. Ciaran wrote this in his final project write up in relation to the up front planning and stocking of his project backlog:

"It showed me the importance of initially understanding a problem. While the process of splitting a problem into small tasks can be quite time consuming and tedious, doing this at a granularity at which I never had to sole responsibility to do before proved to pay dividends as the project progressed. This exposed and opened new and previously unthought of avenues throughout the project. Overall this increased the project's value and when it came time to developing these tasks there was zero confusion allowing the completion of these tasks happen in a fluid and consistent manner."

Ciaran had some strong reflections on the process:

"Numerous times throughout this project I questioned was this the correct methodology to suit a Final Year Project (FYP). Now retrospectively looking back I feel it was the correct choice, while my arguments against scrum revolved around it not being dynamic enough, as college circumstance change so quickly, it was often hard to adapt sprints to suit. While this is a valid argument, I feel the commitment of Small Scale Scrum is needed for a project like this, in that as college circumstances change you are obligated to finish all tasks in a current sprint. This ensures work gets completed regardless, and progress is consistently made. I feel being more dynamic would lead to less completed work with added pressure on the final deadline."

The comments about being dynamic and flexible are something that Leigh is going to use to continue the adaptive nature of evolving Small Scale Scrum and in part has contributed to our thoughts on the Sprint duration being possibly as low as days versus a full week.

2.4 What We Learned

For Leigh, as someone who works in the coaching mindset area, the journey through Small Scale Scrum was incredibly challenging. It really highlighted the gaps in thinking and mindset that would ordinarily go amiss in a larger team, with a longer runway to work with. The time pressure combined with a finite number of people to work on the project magnified issues that larger teams would bat off. For example, having the luxury of an already established test bed and infrastructure were not present in any of the projects we worked on. They became key requirements to try and bring forward and interspersed with the real customer requirements and it was a battle of wills to dismiss them as implementation details when in reality they would help dictate the success of the project. The observations of how our Agilists faired while put into a more confined team have confirmed to me that knowledge of the Scrum Guide and the overall guiding principles of Agile and Scrum are simply not known by the teams that practice that way of working. I feel that teams get the mechanics right for the most part and the improvements seen have moved the team forward far beyond where they began. Teams have gotten comfortable and accepted changes and limitations that ultimately change what Scrum, the framework, purports. Running a team without an available Product Owner, accepting stories that are not ready to be consumed, and even accepting and finishing stories that were simply not complete were just some examples that I observed. Modified scrum has gained on 'pure' Scrum in each state of Agile report over the last few years. I'm now questioning whether the modifications are really modifications or are they simply an abandoning of the principles and guidance the Scrum Guide provides? Are they a cheap way out instead of tackling a challenge? This has motivated me to spend more time in the Small Scale Scrum world and continue to evolve the model and gain more of an understanding into why deviances happen to the guidelines issued.

For Agnieszka, being hands on Software Developer and Quality Engineer at the same time as an MSc Student investigating and implementing Small Scale Scrum was demanding but rewarding. Engineers are passionate about problem solving and she was excited to take on the challenge of finding new ways of working for small teams in small consulting engagements. Her research, found her yet again performing an uneasy multirole act. She observed that context switching proved to be the most difficult, challenging and time consuming for her and for small teams in general. Too many uncertainties from the very start of the project, dynamically changing, unclear requirements and varying personalities, both team members and customers could result in project failure, if not managed timely and respectively. Having team members without overlapping roles was very positive in this case and having the customer fully onboard with a change from Waterfall to Scrum-like process was significant and key to the creation of Small Scale Scrum. Having projects with varying number of team members and different scope proved difficult from statistics point of view and made drawing conclusions challenging. Unfortunately every project is different and finding the most similar projects was difficult. Having access to more small size projects for different customers would benefit and strengthen the framework. On the positive note, encouraging the entire team to participate in the creation of the Small Scale Scrum framework in line with Red Hat Open Decision Framework (ODF) through surveys, interviews, random chats ensured that the framework was truly created "for the team and by the team". This was critical to make the framework a success. Today, the benefits of Scrum and Small Scale Scrum are openly and transparently acknowledged in Red Hat Consulting through Red Hat trainings and offerings such as Red Hat Open Innovation Labs integrating and sharing some of the practices and tools mentioned in the Red Hat Open Practice Library.

One of the more intangible results of executing Scrum in small teams is the mentality of Agile that emerged. Often in a regular sized team, you would have a small number of Agile Champions within the team. That is to say, you have people who have not just bought into the process of adhering to Scrum and Agile principles, but whom project that onto the team. These champions often mask the lack of Agility in the rest of the team as they tend to lead initiatives and be more vocal. From our experience, this means that teams who practice a form of Agile are often mechanically Agile. Those teams and individuals are able to follow ceremonies, they know how to story point, know how to break down tasks, are able to hold Sprint Planning and Reviews and the mechanics of Scrum or Kanban are more or less textbook implementations. The bulk of a team fall into that follower mode, where they are able to be a participant and in some cases drive it. However, the reasons why we perform these ceremonies are often lost. The values and principles behind them are practically invisible to the individuals practicing them. In Red Hat, our Scrum Teams who later branched out to try and experience other Agile approaches such as Kanban, really struggled to grasp the concepts. The muscle memory that came with following Scrum only extended to the mechanics versus the principles. This really shocked Leigh, as a coach of the team who had worked extensively with them for almost 3 years. When presented with Small Scale Scrum, the members who worked with students and whom ran mini internal projects in this manner got a much more rounded experience of Scrum and the Agile principles that underpin it. When brought back into their teams, the change in approach was noticeable but in an intangible way. They challenged assertions around how we tested and looked at Quality. They actively pushed back on the Product Owner for having incomplete stories flowing into the team and rigorously enforced a DoR that ultimately protected the team. A noticeable spike in communications came from the team with a noisy environment being generated that ultimately had to be addressed because it went too far into the over communicating sphere. It did however showcase that a happy medium could be found for the teams to radiate relevant and timely information. While the individuals had that growth mindset, they propagated it at a team level by acting as additional Agile Champions. The team around them became much more self-organising with decisions being made with discussions that centered around the 3 Pillars of Scrum and the values behind Scrum. In the 3 years of coaching the team, conversations that centered around those topics were infrequent. The team, through the empowered individuals who undertook Small Scale Scrum projects, now had a stronger understanding of why we make certain decisions and why certain actions are taken. In one of our longer running Engineering teams, the ScrumMaster found a remarkable change in the team. They no longer required as much coaching in the day-to-day sense that she previously would have found herself engaged in. The team became a lot more self-organisaing and aware of their communications and their actions to the point that the team's reliance on their ScrumMaster lessened. She found herself with a lot more free time to focus on other teams that needed her help more and this really helped us benchmark where our Agile teams should be from a maturity perspective.

3. ACKNOWLEDGEMENTS

We would like to thank our colleagues and peers in Red Hat for their support and patience with us while we asked questions and matured our work. Our students, in particular Ciaran, and lecturers in Waterford Institute of Technology played an equally key role. Rebecca, thank you for your time and patience as we juggled our day–to-day work and took on your amazing feedback, we are indebted to your guidance in making this report a possibility.

REFERENCES

Gancarczyk, Agnieszka. "Small Scale Scrum: A Framework for Successful Implementation of the Scrum Methodology for Small Sized Teams". MSc thesis. Waterford Institute of Technology, Waterford, Ireland. 2018.