



Merkur and LeSS - A Success Story

Owner: Wolfgang Richter

Companies Involved: JIPP.IT GmbH

Content

[1 Introduction](#)

[2 Context & Customer](#)

[3 Pre-Agile - How it all began](#)

[4 Single-Team Scrum](#)

[5 Pre-LeSS – Slightly Scaling, LeSS inspired](#)

[5.1 Why waiting with the LeSS adoption?](#)

[5.2 Preparations](#)

[5.3 Early pre-adoption steps](#)

[6 Application of the LeSS Framework - The beginning of the success story](#)

[Preparation](#)

[6.1 Who - Organizational Structure](#)

[6.2 Who – Line Managers](#)

[6.3 Who – Scrum Masters](#)

[6.4 Who – Product Owner](#)

[6.5 Who – Feature Teams](#)

[6.6 Who – Stakeholders, Users](#)

[6.7 What - The Product \(or Service?\)](#)

[6.8 What/How - Product Backlog](#)

[6.9 How - Backlog Organization](#)

[6.10 How – Sorting/Prioritization](#)

[6.11 What/How – What is Product Ownership](#)

[6.12 How – Approval Process](#)

[6.13 How – Reporting](#)

[6.14 How - Sprints](#)

[6.14.1 Sprint Planning 1](#)

[6.14.2 Sprint Planning 2](#)

[6.14.3 Daily Scrum](#)

[6.14.4 Coordination](#)

[6.14.5 Sprint Review](#)

[6.14.6 Sprint Retrospective](#)

[7 Development Practices / Tools](#)

[7.1 Definition of Done](#)

[7.2 Test Driven Development](#)

[8 Communities](#)

[9 Summary](#)

1 Introduction

This case study reflects on the Large-Scale Scrum journey of an Austrian insurance company. Its environment was pretty stalled and not used to change quickly or being a learning organization. Hence, it faced challenges and incidents typical for adopting the agile mindset in such an environment.

The journey towards LeSS passed four phases:

- 1.) Pre-Agile
- 2.) Single-Team Scrum
- 3.) Pre-LeSS (slightly scaling)
- 4.) LeSS

In this case study the focus is on phase 4 – the LeSS adoption itself. The first three phases will be described only briefly to understand the motivation, and the reason for choosing LeSS as a framework.

2 Context & Customer

The Merkur Gesundheitsversicherung is one of Austria's largest insurance companies.

About employees 50 work at Merkur IT, the main focus of this case-study. One of the main responsibilities of Merkur IT is to ensure a trouble-free infrastructure, individual software development for the larger organization and support various heterogeneous applications needed to perform the daily business.

3 Pre-Agile - How it all began

The journey to agility for Merkur IT started in 2012. A legacy system was intended to be replaced. The system in question had a history of several preceding replacement attempts, and all failed. All previous replacement endeavours relied on traditional plan driven approaches based on sequential phases, specifications, and milestones. After some research and with the help of JIPP.IT an experiment with Scrum as software development framework was started. Scaling was not wanted at this time.

4 Single-Team Scrum

In the early Scrum adoption phase a single team was set up. Agility was not welcomed by everyone, yet. For most people involved it was a major change of their work habits, since people were used to work in plan driven methods, in some cases for over 25 years. Another aspect to be considered was the company existed for over 220 years. Some of the structures, and processes felt *old*.

The first phase of the journey was a huge success, even with only one team applying Scrum. The replacement of the legacy system took two years. That might not sound like a major success, but the preceding multiple failed attempts at replacing this system spanned a period of 8 years! Hence the successful replacement was considered a successful use of Scrum.

Motivated by this experience Merkur IT management decided to be ready for the next bigger change.

5 Pre-LeSS – Slightly Scaling, LeSS inspired

5.1 Why waiting with the LeSS adoption?

In this period, the company started to reinvent itself. Not only scaling of Scrum at Merkur IT was potentially considered to be a valuable decision, but also other areas of the Merkur group aimed for change. The following list shows a few of these endeavours, and their influence to Merkur IT:

- Changes in some management functions had influence to the planned endeavour.
- A rebranding campaign, from a typical insurance company, to a “Vorsicherung” (preventive medicine, and activities) was launched. This had potential impact to the product definition.
- The market image was planned to be changed, to become more modern. More transparency, but it had to be decided what to make transparent.

This phase must not be understood as the actual LeSS adoption, but as a period used for some preparations for the later LeSS adoption. It was definitely LeSS inspired, but too many implications prevented a real LeSS adoption in this period. However, guiding principles already considered were e.g.:

- Narrow & deep
- Top down & bottom up
- Volunteering

From a coaching perspective, insights from various disciplines, such as occupational psychology, behavioural psychology, sociology, economics, and other disciplines rounded up the toolbox already in this phase. Later they became essential tools for the real LeSS adoption.

5.2 Preparations

The first step towards the adoption of LeSS was a strategy workshop with the new head of Merkur IT and the head of software development, the two main responsible line managers. Interest to apply a (de-)scaling framework did already exist. Also LeSS to be the framework of choice was already considered very likely, but the two managers wanted to analyse it further, so their decision can be underpinned with some facts, and figures.

The result of the workshop was intriguing: In their eyes, LeSS was definitely the most compatible approach to start scaling agility at Merkur IT. In succession their main goals for the adoption were set and the reasoning behind scrutinized. Early adoption moments were sketched, and the expected improvements depicted. Overall, the starting point became very clear.

Their main goals for the adoption were:

- More flexibility -> later understood as an aspect of agility
- More focus -> also understood as part of agility; side-effects: reducing multitasking, reducing too long cycle times
- A common picture -> later understood as “whole product focus”, including understanding the entire product release, focussing on delivering Increments every Sprint, engagement of everybody involved;
- A better way of prioritising requirements -> resulted later in one Product Backlog
- Better communication within and between teams, and other departments in the larger organization to achieve the highest customer value -> resulted in faster feedback cycles, more transparency, supported by LeSS events and principles as walking-aid
- Quality (as a more abstract definition) - Quality was mentioned in several contexts, e.g. requirements definition, communication with stakeholders and users, code quality, documentation, planning;

Also more responsibility of teams was mentioned, and more understanding of requirements in the teams, but there were not identified as the main goals.

Some sketches of the workshop, with different approaches, and brainstorming results, leading to the above mentioned goals were:

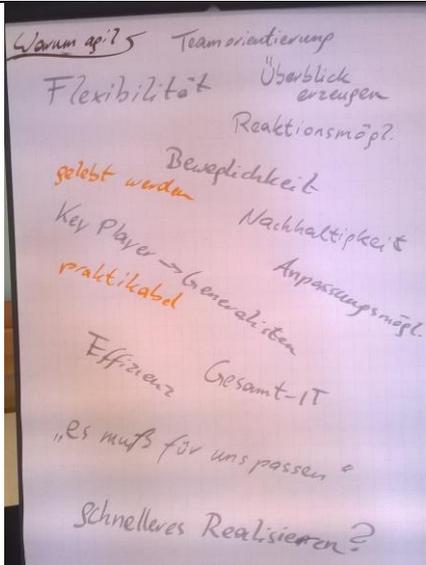
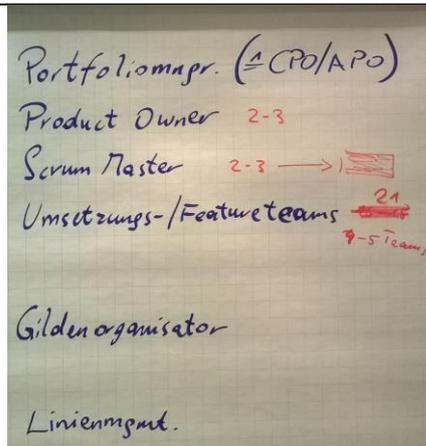
Description	Image	Notes
<p>Image 1 - Why does Merkur IT want to be agile</p>		<p>Goals like flexibility, shorter cycle times, etc. were identified. It was mainly an identification exercise of the main existing pain-points, and a retrospective of the single-team Scrum experience prior to LeSS.</p>

Image 2 - Roles (LeSS or not LeSS, LeSS or LeSS Huge)

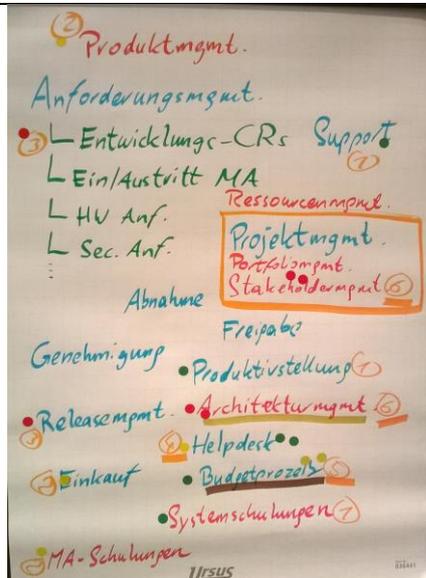


On this flipchart it can be seen, that the understanding of roles was not clear, yet. Especially the question, what a Product Owner is supposed to do and what not (like clarification) played an important role in this exercise.

Also whether one "Requirement Area" or more than one are desirable, was figured out experimentally.

The final result was: One Product Owner, one Product, one Product Backlog, and LeSS (not LeSS Huge) as framework of choice.

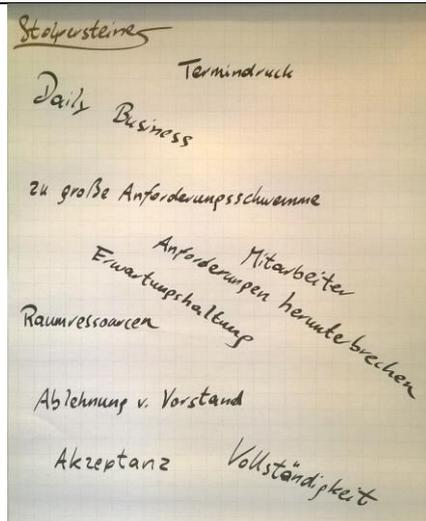
Image 3 - Requirements, Product Management, Order and Approval



This flipchart reflects the result of the process identification exercise. It shows a status-quo overview of different types of requests (requirements) for Merkur IT, existing positions involved (stakeholders), existing processes involved.

Most of these processes were supposed to be simplified with LeSS, unfortunately not all of them, as a few are predicted by law. However, it became clear, the potential for simplification is significant.

Image 5 - Potential Pitfalls of the Adoption



The participants wanted an overview of potential risks for the adoption. This flipchart shows the most relevant risks, leading potentially to a failed adoption, or a significant delay.

Risks identified were e.g.: Time pressure, too many requirements at once, expectations of stakeholders, and infrastructure amongst others.

Those risks were addressed later in other workshops with stakeholders, team-members, etc.,

We did not use Causal Loop Diagrams in this workshop, as the participants had never worked before with this tool. Hence we decided to use brainstorming, sketching, cause-and-effect descriptions, and other better known tools for analysis.

Some discussions were rather emotional, e.g. about portfolio-management, how to support releases, and who else other than a Product Owner is supposed to support these aspects. As a result, it became clear, there were no dedicated management positions, such as Release Manager, or Portfolio Manager required.

Also the question about requirements clarification, the role of Business Analysts, and potentially one PO per team was addressed. It became clear, that the desired situation is clarification is done by the teams themselves, hence they needed skills for business analysis, but no dedicated “PO-per-Team=BA” role.

The strategy workshop was followed by others with more people involved. In these workshops details about processes of the larger organization where explored. Some of these processes were very difficult to change from the inside of Merkur IT. Hence, support from other areas of the group became relevant for the upcoming adoption. Also alternatives to changing these processes were elaborated, in case a change cannot be initiated, yet.

5.3 Early pre-adoption steps

Meanwhile two teams started to work with Scrum. There was no proper Product Owner, yet, but the line-manager together with the “Product Owner” of the first Scrum Team took over activities such as sorting of backlog items, strategic planning, and some communication with stakeholders. It was a Fake-Product Owner situation.

To start, each team had their own so-called Product Backlog, which of course was actually just a team backlog rather than true single Product Backlog shared by all teams. Naturally, this led to local optimization problems, difficulty in seeing the whole perspective, and inflexibility (anti-agility) of a team to work on items in the other team’s backlog.

Clarification was done by either team members, separate business analysts, and/or the line-manager. Prioritization was done sometimes independently, informally in meetings between the various responsible authorities, and in tools. This was obviously a situation that needed to change soon.

During this time a new employee was hired, who was supposed to take over the (one and only) Product Owner role later. As this person was new to the company, the product, and the role, he required some learning curve and was not able to perform the role of the Product Owner, yet. Hence, the Fake-Product Owner situation continued for about 4 months. This period faced an additional delay, since personnel at relevant positions in the organization changed, too.

Main challenges / key moments:

- Workshop to figure out whether LeSS is the framework of choice for the organization – Result: Yes.
- No (real) Product Owner available – Fake-PO situation to pass this phase, and allow learning for the new employee.
- Distributed planning/prioritization

6 Application of the LeSS Framework - The beginning of the success story

Preparation

After LeSS was chosen as an appropriate scaling framework, or perhaps better described as our chosen *de-scaling* or *simplification* framework, a date was set to start with the first concurrent Sprint. But before that some questions had to be answered:

1. Who - Organizational Structure
2. Who – Line Managers
3. Who – Scrum Masters
4. Who – Product Owner
5. Who – Feature Teams
6. Who – Stakeholders, Users
7. What - The Product (or Service?)
8. What/How - Product Backlog
9. How - Backlog Organization
10. How – Sorting/Prioritization
11. What/How – What is Product Ownership
12. How – Approval Process
13. How – Reporting
14. How - Sprints

Some of the processes, which existed for many years, had to be adjusted slowly. Otherwise legal implications and potentially existence-threatening mistakes could be overseen. Accurateness had to remain the main constraint and concern. Especially in areas involving financial market authority. Hence, process changes in these areas went through several authorities first for plausibility checks, and approval.

Also those people (such as other developers) not involved in the previous single-team Scrum experience, without any previous agile or Scrum trainings had to be brought on board. Trainings were scheduled, info-material was distributed, and reasons, goals, and the vision was communicated

6.1 Who - Organizational Structure

Prior to adopting LeSS, the existing organizational structure was rather simple already, comprising the head of Merkur IT and three department heads. Additionally, Team Leaders were supposed to support line management with appraisal interviews and senior expertise but did not have authority over people.

The bigger challenge was to find an acceptable way of how to build teams, how to enable them for Large-Scale Scrum, and how to integrate operations people.

When adopting LeSS, one of the main interest is to ensure that the organizational structure does not prohibit responding to change. Therefore, the focus was set on Feature Teams, the flow of requirements through the company and working on a single Product Backlog sorted by one Product Owner.

The first adoption phase of LeSS was to start with development oriented staff, and to add “pure” operations staff to the teams later. As can be seen in the diagram below, staff involved in IT Operations remained separate in the first phase. It was supposed to integrate them into Feature Teams as a second step. This decision was made due to surrounding changes in the overall organization, which required more input from other people. In order to continue with the adoption without significant delay, this compromise had to be made, and as written earlier, is not fully resolved.

The department comprising some “PMs”**, and other cross-departmental functions, such as secretary, remained separate, without another dedicated line manager. The senior manager role for this group was assigned to the head of the business unit.

** “PMs” stands for “Project Managers”, but must not be understood as the traditional role description. PMs in this context are mainly stakeholders, and internal clients. This term is used in the Merkur group to identify people who can represent the board of directors for certain endeavours. It may be confusing for readers not involved in this adoption, and especially native English speakers, that this is not the traditional meaning. In German sometimes English terms, such as “Project Manager” are used in different context, and as placeholder for something else. However, this part of the organizational structure holds further potential for optimization.

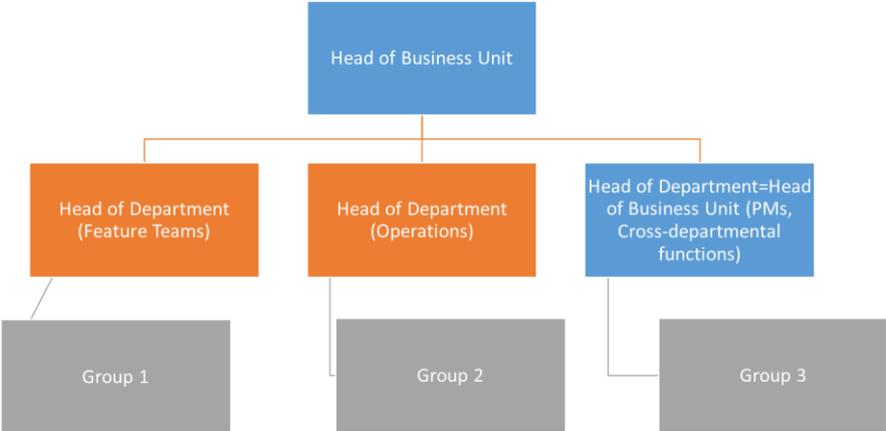


Image 5 - Organizational Structure

Group 3 can basically be seen as a group of business analysts and project managers, and also home for other functions outside or in addition to feature teams, such as Scrum Masters, and the Product Owner. This is consistent with LeSS (and LeSS Huge), as those roles are seen as a peers to the developers, instead of being hierarchically “above” the developers. For the interested reader, please refer to “Large-Scale Scrum: More with Less” (Larman/Vodde, © 2016; ISBN-10: 0321985710) on page 99 for the why.

About two months after the first Sprint, the involvement of IT Operations began. This part of the journey still continued, as operations was not fully included in the new structure, yet.

6.2 Who – Line Managers

The role of the line management changed recognizably. Before the transition line-managers were deeply involved in projects. They took care of prioritization of projects, of requirements, estimated projects, assigned projects/requirements/tasks to people, were involved in clarification, and took care basically of all project management activities. And *product* management was basically done by the line-managers, too.

As written earlier, before the actual LeSS adoption, due to a lack of available people, a “fake-Product Owner” situation existed in the beginning. This meant, one of the line-managers was still involved in Product Backlog sorting, and even clarifications.

However, line-management had to change and became merely a combination of enablers, consultants, and - as far as possible - organizational coaches. Due to Merkur group policies they remained the main authority for organizational decisions, which - seen from a Scrum and LeSS perspective - can be a risk to the empowerment of Feature Team members. In a worst-case scenario this could be understood as results of retrospectives for organizational change would always need the approval of the line-managers. To a certain degree this was even correct, as the knowledge of e.g. legal or economical implications of organizational changes was with the two mentioned line-managers. However, the two line-managers were eager to support the Feature Teams in this regard as much as possible, are available for developers, Scrum Masters, and others for Q&A about potential legal and economical implications of organizational changes, and continued to improve the organizational design together with the developers, the Product Owner, Scrum Masters, coaches, and stakeholders.

They adopted techniques like go-see, self-management, teaching problem solving, coaching and mentoring. Especially “coaching” was a critical activity, as the term comprises certain experiences, educations, and skills. Coaching of the line-managers refers to supporting the Product Owner, Scrum Masters, and developers, as well as stakeholders in the Merkur group with their knowledge about the organization, about existing rules and policies, about development techniques, and domains. A goal often repeated by the head of development (one of these two line-managers) was, that the Feature Teams become more and more self-managed. In this regard he sought support of the Scrum Masters, but also fed back his knowledge about working with colleagues, different team setups, different process-landscapes over the last twenty years within Merkur IT.

A particular example concerning the actual LeSS adoption was, that he acted as a customer in one of the meetings, and asked critical and powerful questions to the developers, so they realized, that certain statements are either risky, or irrelevant for the clients. Developers tended often to explain the technological background of features, but forgot to mention the actual benefits to clients, and what implications the new features might have to client’s day-to-day businesses. So he initiated thinking processes in this particular situation, made developers think more about the *why* and how they are connected to other areas, or processes from a client or customer perspective. He also asked about his (the client’s) role in the development itself, and what was expected of him (the client). Developers then replied, “I need potentially certain clarification”, “maybe you need to talk to other stakeholders, in case there might be conflicting views on outcomes”, and “I need within the next days some testing, and an acceptance”. In the beginning this was for some of the developers rather unusual, as such direct conversation with stakeholders was often done by someone else.

In another situation the head of IT simply watched a review meeting, took notes, to talk later to the Product Owner and the Scrum Masters about his observations. Wording used was more in a question-style, than in telling them to change certain elements.

This was a learning process, as figuring out what to ask, what to present, and how to interact requires practice. Allowing to practice, and risking potentially complaints from stakeholders about communication not going smoothly, but with too many people involved, too little outcome - which happened - cannot be taken for granted right away, but has to be enabled. Both line-managers knew and accepted this risk, and accepted the complaints of stakeholders, but together with the Product Owner, developers, and the Scrum Masters found right measures to improve this. E.g. information meetings about the new LeSS approach were held several times for stakeholders, people from outside Merkur IT were asked to give feedback to their interactions, and information about their concerns, advice from developers was taken seriously and fed into the improvement activities.

A challenge for both line-managers was to switch between coaching techniques, and consulting. The main activities could be more seen as consulting, and potentially mentoring. But within the first six months of the adoption, certain behaviours changed from consulting towards coaching. E.g. asking powerful questions, taking empirical journeys, opening doors for learning elements, and for change are typical coaching techniques and activities. Especially helping to establish the four quadrants relevant for change (knowledge - motivation - empowerment - ability) were strongly in their focus.

Concerning *knowledge* they shared their knowledge, and created leeways for people to have time and space to learn. Concerning *motivation*, they enabled and created environments, facilities, and supported the creation of new structures and policies together with all involved to be able to grow intrinsic motivation.

Concerning *empowerment* they removed e.g. too complex processes for the approval of features, and simplified structures and processes to allow people to contribute to continuous improvement. Concerning *ability* they e.g. hired coaches for the adoption, supported hiring of people for Feature Teams, in which cross-functionality was not possible, yet, or enabled people to learn and improve communication with stakeholders. Just to mention a few examples of their long list of supportive activities, which often goes hand-in-hand with Scrum Master activities. How the line-managers and Scrum Masters will interact, or overlap in future, will be seen.

Additionally, the idea of servant leadership was formally introduced. An in-house workshop for this specific topic was held. Amongst the participants were not only relevant line-managers, but also other team members, and one of the group Project Managers.

As both line-managers had a lot of insight into the wider Merkur group mechanisms, which was not directly visible to developers they often remain an interface to Merkur group structures and policies. This was potentially one of the less LeSS-like elements of the adoption. However, changing the entire Merkur group was not in scope for this case-study and/or adoption.

Summarized can be said, responsibilities, and activities of the line-management changed to:

- Organizational Development - together with the support of coaches
- Simplification of structures and processes
- Support of team- and personal development - together with the Scrum Masters
- Improving the working environment, so teams can work effectively
- Support in organizational and individual problem resolution
- Authority for staff such as hiring and firing - (might change in future)
- Knowing the goal of the organization and communicating it to the external parties
- Supports in case of negotiations with suppliers
- Is primary contact for institutions, and partners outside, like institutions for research, academic institutes, networks, etc.
- Thinks like a coach

- Senior Management:
 - Responsible for legal aspects, is liable from a legal perspective for the organization
 - Coaching aspects are more from a helicopter view at this management level (coaching in terms of helping to understand things which are not directly visible or accessible)

As can be seen some of these activities overlap with Scrum Master activities. There was a potential risk of conflict due to parallel or invisible, even contradictory activities. In the beginning such things happened, such as teams got different information about how to progress with tasks in case stakeholder interaction is required. This was later resolved by a formal meeting installed between the Scrum Masters, the line-managers, and the Product Owner. It was basically a “Weekly Sit-Down” between them. It helped to synchronize better. Whether this meeting will be kept, was not clear, yet, but it turned out to be valuable at least in the first couple of months.

Therefore, partially the goal for line managers was similar to the goal of a Scrum Master, namely organizational development, simplification of structures and processes, coaching and mentorship of people. Additionally, training, and supporting with experience, got predominant. Both line-managers had a large set of experience at Merkur, hence they were the right people to ask specifics and peculiarities about domains, tools, regulations, etc. The head of development also had good insight into the architecture of the many heterogeneous systems, hence could be consulted for certain clarifications.

From the outside it was very refreshing to see, that these line-managers enabled transparency, encouraged people to learn and innovate, and were available for questions, or support in case of e.g. felt dead-ends, etc.. Whether it was about new requirements in terms of tooling, escalations due to unavailable stakeholders, emergencies in terms of missing tool-chains or equipment, they supported the teams. The mentioned overlap to activities of Scrum Masters continued due to legal constraints. Not all activities were possible to be completed by Scrum Masters in this environment, yet. The line-managers were also mentors and enablers for the Scrum Masters, as they still needed to learn. It looked like a very good balance of management, mentoring, and support, but with still potential for further simplification. Maybe in some time they can switch to become Scrum Masters, or Feature Team members. The only road-block for that were group policies, such as power of attorney only assigned to an explicit management function.

The line hierarchy itself was not supposed to be changed in the LeSS adoption, since it was already simple. Only the term “Team Leader” was removed, but this position was not shown in the line hierarchy anyway. However, this change was a challenge. Most seemed to accept this title or position change pretty well. However, it had no meaningful impact overall, but had to be observed.

6.3 Who – Scrum Masters

During the Single-Team Scrum experience, and the Pre-LeSS experience, two Scrum Masters emerged. One internal employee, one contractor. They were supposed to continue with their role. The biggest change was, that before LeSS the internal Scrum Master was only a part-time Scrum Master, and a part-time developer. He was considered the only expert at Merkur IT for a specific technology-domain combination. Obviously a knowledge silo without redundancy.

Prior to Scrum and LeSS such situations were accepted. Already with Scrum, and later in the preparation for LeSS such knowledge silo situations were deemed to be unacceptable. Hence other employees were educated to become knowledgeable enough to allow this person to become a full-time Scrum Master. Learning mechanisms used were e.g. Q&A workshops, pair programming, pair-

writing, and specification by example. As a result the Scrum Master was able to become a full-time Scrum Master. Mentorship for the newly educated people continued, in case it was required. But the amount of such support requests gradually decreased over time due to a good established knowledge exchange scheme.

Prior to LeSS the contractor so-called Scrum Master also unfortunately acted as a separate Business Analyst, clarifying Product Backlog items, creating specifications, and handing them off to the team. Of course this created many wastes as a result. At this time, the group had the widespread misunderstanding that Scrum contained “stories” rather than “items”, and furthermore, the widespread misunderstanding that “stories” meant writing them in some template style (“e.g. who-what-why”). It took some time for that misunderstanding to be replaced by the original idea of *stories*: that the real *coders* and real *customers talk* regularly face-to-face to clarify and understand together.

But when the understanding did start to emerge, and therefore the teams took over the responsibility of clarification (“telling stories with customers”), then this contractor Scrum Master was freed up from that task, and had the potential to focus full-time on being a “real” Scrum Master and coach, instead of a mixture of a Scrum Master and business analyst.

A less-optimal situation was the two Scrum Masters/five Feature Teams ratio. Everything worked pretty well, but some team-members started to think about having more Scrum Masters in place. In certain situation it happened Scrum Masters were occupied in one scenario, happening in one particular team, but would have been great support in another. The Scrum Masters themselves felt like being overwhelmed sometimes with too many important tasks on their todo lists, and, hence, not being able to serve the teams well enough. Especially when everything was new to many team-members - Scrum, LeSS, self-organization, transparency, etc. - Scrum Masters were required to observe, listen, and coach many at once. A one Scrum Master/one Feature Team situation would have been welcome. At time of writing of this case-study, it was listed as an impediment and to be resolved by line management.

6.4 Who – Product Owner

As mentioned, for a certain period a Temporary Fake Product Owner (TF-POs) situation existed. Eventually a new employee was hired to become a true Product Owner who was a real business-side “owner of the product” with independent authority. The new employee had to learn about the product, the organization, and everything in between, so it required a learning curve for this person first. Learning was supported by activities such as:

1. Info material – a lot of reading
2. Interaction with clients together with the TF-POs, and people from Feature Teams
3. Using the various systems comprising the product
4. Workshops with internal and external coaches about the processes (LeSS, and traditional Merkur group)
5. Information exchange with the Scrum Masters about processes, the pre-LeSS/single-team Scrum experience, existing and retired processes
6. Simulations of Product Backlog prioritization activities, and planning games for releases, and doing road maps

As said a newly hired PO was not an ideal situation during this adoption. However, even with the PO as sole authority for the product, product ownership required team-work. Inputs given by experienced people, stakeholders, clients, and others were very important and mandatory for the

new PO to become more and more familiar with his environment. However, it showed also, that the newly hired PO cannot gain the same experience about the product within a couple of months, as someone, who knows it for years. But the situation was also seen as a chance, that the new PO has the guts to break certain existing patterns.

As the new PO had broad industry experience, the expectation was that he may come up with new ideas, or new parameters for value identification. This turned out to be a feasible expectation, as he started to open doors for new market image campaigns, and together with line management and stakeholders of the Merkur group allowed new aspects of a product definition to be considered. So to say a breath of fresh air breezed through Merkur IT!

After the new hired Product Owner was ready enough to take over his responsibility, the temporary fake POs situation was eliminated. Now only one sole authority for this role existed.

In the beginning (the first three Sprints), the new Product Owner still spent significant time on clarification, which was not supposed to be done by him, nor was he able to do clarification for all teams, and requirements. He realized this misconception soon. Feature team members were trained to do clarification within the teams. Workshops about, how to extract the “real requirements”, how to eliminate wishful thinking expectations, how to read between the lines, etc. were held. The theory was covered in workshops, and practice mentored in day-to-day business by the Scrum Masters, experienced colleagues, and an external coach (myself).

Not all teams welcomed this new set of tasks. Teams were asked to take over clarification tasks gradually, also to allow the PO to fulfil his actual responsibilities. Especially one team seemed to remain reluctant to take care of clarification directly. The reason for this reluctance turned out to be caused by unplannable changes in team setup, such as the death of an experienced team member, parental leave of another, and a remaining team comprising one experienced employee together with apprentices and new hires. The situation was not ideal, but it improved significantly, and mentorship, trainings, as well as “pair-writing” of items continued. All teams learned about the importance of clarification, understanding, and interaction with clients, and stakeholders. It was great to see, that people asked for better acceptance of time invested in these tasks.

Another challenge with regards to clarification was the existing organization. People in the wider organization were used to have one main authority accompanying clarification meetings. They expected the PO to be present in any of these meetings. He would have needed to be in too many meetings for this purpose, hence training to change these expectations in the wider organization were required. As said, some processes and structures and habits were as old as the company itself. Those things were not easy to change, and still were not fully changed. Some of them will require some more years to change, some potentially will never change.

6.5 Who – Feature Teams

Before LeSS an organizational split between Java and mainframe developers existed. It was basically a component-team situation, with development areas, and additionally changing affiliations. This means people were assigned to teams based on projects. Once a project was over, team-members were moved to potentially new teams. Or worked independently on a new endeavour.

It was very likely that Java developers and mainframe developers together had to deliver one part of the solution in order to produce a feature of customer value. Before LeSS a huge investment into communication, coordination and detailed upfront planning was necessary to accomplish an on-time-delivery of each part. To minimize this overhead and delay, and to simplify the organization, the goal was to change team structures to Feature Teams.

In the Pre-LeSS phase, a *Feature Team finding* workshop was organized. Inside given conditions people were allowed to find their own team. Five teams of nearly equal size were formed. As one of the first team-building activities the teams were asked to find team-names. No limitations were given; teams were allowed to come up with any kind of name. The exercise went well and teams seemed to enjoy this. They soon started to identify themselves with these names.

To support the team-identification even further, collocation was supported by line-management. Under given constraints of facilities, the teams moved to new offices, with more modern infrastructure, new table-settings, whiteboards, flip-charts, and enough space to sit or stand together for cooperation. This was pretty much the first step by the new line-management team to support the new paradigm.

But the most significant change was, that teams were now cross-functional, and able to work cross-component in a “full stack” to create end-to-end features. In the first few Sprints, knowledge silos, and component thinking was still very strong. In certain cases the problem was different clients requested different features, required to be implemented in various different systems with little to none interdependency. At least from a developer perspective. From a customer perspective, there were at least sometimes some interdependencies, but even seen from the customer point of view, certain requirements did not have obvious cross-relations. So the question came up, how to identify these cross-relations, if they are not tangible? And how to move away from component thinking towards a system-wide product view?

Example: Department A requested a change in system 1. Department B requested a change in system 2. Both systems comprised features required in department A and B. However, the requests did not seem to interfere or have interdependencies. As department A did not know what department B requested, and vice versa, they were not urged to talk to each other .

From a developer standpoint the requests also did not seem to have interdependencies, as they were meant for different systems. But: From an end-to-end perspective they may have had side-effects to each other, since the end-to-end process changed across systems, and, hence may cause changes to day-to-day business of both departments.

To identify such cross-relations knowledge about the end-to-end process was required, but in certain cases only very few people did have this insight. If they were not informed, these interdependencies could have been easily overseen.

These sometimes hard to grasp end-to-end processes were one of the reasons in the beginning, why Feature Team members seemed to continue to look at the product from a developer, or system-centric perspective. The customer centric perspective was wanted by the PO and line management, but due to a rather complex and heterogeneous system-landscape, and additionally many domains, this was hard to achieve, and still is. This had also to do with the new product definition. In the past systems were seen as synonyms for products. With LeSS a new, more customer centric view was introduced.

But as said, to understand this paradigm shift in theory and practice, and to allow also the clients to understand and accept their interdependencies across domains, it required more time. One of the line managers, who had a pretty solid understanding of these cross-relations supported and consulted both sides, developers and clients.

The evolution of the Feature Teams continued. In some teams knowledge-silos still existed, but it was difficult to share knowledge collected over years or decades in a couple of months, or a year. Important was, that this process started, and silos became less silos, but more and more regions with

better access for others. Border crossing became allowed. This lack of shared knowledge was still an ongoing challenge for the teams, but it improved by the week. Line management supported and expected knowledge sharing.

Drawbacks from time-to-time were changes in the Product Backlog due to e.g. very late reported regulatory changes, legal changes, and other unplanned requirements from the outside. In the past individual team-members got such “Querschüsse” (“a spanner in the works”) assigned, but with LeSS these short-notice requirements were added to the Product Backlog and treated as any other item. Urgency and other parameters, such as potential business implications were treated as influential attributes for their business value, and all these attributes were input to frequent re-prioritization based on learning each Sprint.

Main challenges / key moments:

- Feature Team finding workshop.
- Collocation.
- Knowledge silos, and “independent” tasks for individuals slowing down team-building.
- Cross-team cooperation happened automatically.
- Better coordination across teams for better Increments.
- Less management overhead required for task assignment, and component integration.

6.6 Who – Stakeholders, Users

Stakeholders already involved in Scrum for the first team gained insight and routine in how to use agile principles, and practices for better outcomes, and simpler collaboration. Stakeholders not yet involved in any agile activities remained sceptical. Especially in the first weeks of LeSS adoption many escalations occurred, since team members replied to requests with ambiguous replies. For example, Stakeholders were (correctly) told that during a Sprint no new requests were allowed. This led to confusion, as prior to Scrum requests were allowed on a daily basis. The new policy and planning approach led to escalation to line-management.

Some stakeholders still were reluctant after the information events, as it sounded to them that “more flexibility” for Merkur IT seemed to mean “we do not get what we want any more”. And of course in certain cases that was potentially a correct interpretation -- people no longer could just expect to get what they wanted on very short notice without analysis of the global impact or influence to other, potentially conflicting requests. Rather, requests had to now be prioritized each Sprint by the Product Owner for business value against a set of other competing items. However, after involving stakeholders more and more in the events, and adding more and more transparency, it began to show the benefits for all groups involved.

Especially with Sprint Planning 1 an experiment took place. It was a bit differently organized as described in the guides.

The first 5 minutes the PO presented the main goals for the Sprint to all Feature Team members. Then the teams selected their items, followed by a short presentation of the main topics per team. They asked other teams for cooperation, if interdependencies were identified, yet.

After that the room was rearranged. All items were grouped by domains. Clients and stakeholders of the various domains were invited. After a short welcome by the PO and a SM representatives of the teams talked to clients about some details, expectations, and how to organize clarification, testing, and acceptance of items. This required team representatives to pair with other team representatives, as the Sprint Backlogs (grouping by team) and the grouping by domains (or customer groups) were

not congruent. In other words the customer-centric view required cross-team and cross-department communication and presentation. This took about 45 to 60 minutes.

After the Sprint Planning 1 was held in this format for the first time, it was the first time team members consistently reported satisfaction. Many insights, such as “clients have the same stressful moments as we do”, occurred. It was really a pleasure to see, what happened there, and how many obstacles became conquerable. And even with a large crowd in one place - which seemed to conflict with meeting policies - it was soon obvious, that this format replaced many other isolated, often inefficient meetings and helped to save much time potentially spent in inefficient meetings. Everyone knew what to expect, what to prepare, and what to take away.

6.7 What - The Product (or Service?)

Before LeSS, there was no clear definition of the Product. In the workshop with the two responsible line managers, the Product was identified as:

IT-innovation, software- and system-services for management and sales of Merkur insurance.

This may not have been the best possible product definition, yet, but it helped to get a different view. Before this workshop “products” was only a synonym for different “systems” or applications. The systems landscape was comprised of ten to fifteen different applications, depending on how one counts it. One preparing-for-LeSS adoption thought was that each system is essentially a product. But soon it became evident this perspective was too limited.

The mission of the Merkur IT was to provide all the necessary infrastructure and software for the different business units of the insurance company needed for their daily business, and systems for their end-customers, such as Point-of-Sales, and Self-Service-Portals. This included the maintenance and the adoption of the current software as well as the development of new applications or enhancements. The system landscape was very heterogeneous. Additionally, domain knowledge was essential. Hence for each of the almost 20 different areas specific subject matter experts existed.

With the introduction of Scrum setup of redundancies and knowledge transfer started. Nevertheless, certain areas require at least 5 to 10 years of experience in order to truly understand the calculations, implications, and potentially expensive consequences for the company, and customers.

The architecture of current systems was mainly monolithic and specialized for the different business units. A main goal of the IT strategy was to move those systems to a more service oriented architecture to be more flexible and make the re-use of those services easier. This was started to be realized in parallel to the LeSS adoption. Larger milestones for the next seven to eight years were included in the roadmap.

Another approach was to look at the services and to see the entire suite of *Merkur IT services* (including services for sales, individual customers, business customers, wholesale customers, etc.) as one product. After some discussions, analysis, and experiments with defining features and theoretical team structures it seemed to be the most feasible approach. The challenge with this approach was to distinguish between operations and development tasks, if these areas were separated in the beginning. Basically the solution was to mark these as absolutely independent tasks. For example, installing a new browser version for employees, if older versions were not supported by the new customer portal were supposed to be “operations tasks”. Also installations of staging environments or database instances remained “operations tasks”. “Leading teams” could then decide to take them on, supported by Product Backlog filtering (a tooling question). Travelers (supporters) from operations could then temporarily join teams, or even support as external experts,

to complete cooperatively endeavours. It might be considered an organizational overhead in terms of time and duration, but it served as a starting-point within given restrictions. The goal was to remove these restrictions over time.

Systems were then internally seen mainly as *tools*. The “service” or “product” changed to what creates improvement for people in the Merkur group, and clients of the organization. The term “value” was used, too, but as this term is more an abstraction of various different parameters, and different potential benefits, it was not emphasized.

A couple of months after this new and broader product definition, people started to think whether an even better product definition might be something around *insurances, tariffs, or something the clients of Merkur can receive*. It became also clear that there was not only one possible view on products, but seen from different angles, and with different context, product definitions can vary. This process of product definition continued, but was consistent with the LeSS rule that a product definition is supposed to be as broad as possible. .

Main challenges / key moments:

- One of the main questions was: What is the product? Is there only one product? Are there many products? From which perspective do we need to define, what the product could be?
- A turn from a system-oriented product definition towards a customer centric product definition.
- Another approach for products was to look at the different insurance areas, e.g. life insurances, health insurances, medical insurances, property insurances, and others. From an end-customer perspective, each tariff would be potentially a product. However, this product perspective still wasn't beneficial enough.
- Distinction between an internal and external view on “product(s)”.
- A paradigm shift for many in the organization, especially those who worked for years with a different view.

6.8 What/How - Product Backlog

Before LeSS, there was no single Product Backlog for all requirements. How to come from distributed requirement repositories to a single Product Backlog, and how to sort items, was a main challenge in the period before LeSS, and even when LeSS started.

One of the main challenges for people in the teams was the handling of requirements. It was decided to note down features mainly in the *who-what-why* format, to create consistency for most items, and that the Feature Teams need to take care of clarification tasks. As for the majority of technicians this was very difficult, people more skilled with requirements elicitation, business analysis, and communication with stakeholders supported the teams. Those people clarified details about user

stories together with team-members, and supported the Product Owner with fine-grained decisions, or other planning activities. These activities supported learning very well. However, for some people understanding requirements, and being able to gather them or writing them down was a huge split.

Team members more and more took over the responsibility of clarifying requirements and adding items to the Product Backlog, but the quality in the first couple of Sprints was as expected, but of course in need of improvement. New mechanisms and tools require a learning curve. However, most team-members were eager to learn, and improve. Some did not want to be responsible for clarification in the beginning, but this was more a symptom to existing patterns caused by almost outdated policies, than general reluctance.

In the beginning the Product Owner, who was rather new to this job, did too much clarification himself, added most items to the Product Backlog, participated in too many clarification meetings, and was simply overwhelmed by the amount of work. He soon found out, that this is not manageable, and started to focus more on the product management, sorting, and deciding aspects. This situation took only about three Sprints to improve significantly.

During workshops questions arose, whether one Product Owner could handle the complexity of different topics, or Area Product Owners should be introduced. After analysing the responsibilities in existing (and currently not changed) processes, it became evident, that one, and only one Product Owner should be responsible for the Merkur IT services. For a LeSS Huge adoption, the organization was not large, or complex enough. Alternatively, the concept of Requirement Areas could have been used also for smaller areas, but it became soon obvious, that the chosen way of applying LeSS principles, and having so-to-say only one requirement area made more sense. To overcome the fear of losing or overlooking important information -- sometimes only existing as tacit knowledge of experienced and seasoned team-members -- subject-matter expert contacts for special topics were introduced. This concept was supportive in the beginning, soon started to change, but still required time to become obsolete.

6.9 How - Backlog Organization

Prior to LeSS, tasks were assigned to teams or individuals by managers. This paradigm already changed when Scrum was introduced for the first team. With five teams the Product Backlog organization and selection mechanism had to be taken to the next level. The goal was, that each Feature Team was able to select from the overall Product Backlog (Merkur IT Backlog). But in case a team chose items very counterproductive repeatedly, the Product Owner was allowed to overrule a team's choice. Usually teams were supposed to learn and to look beyond their own nose. However, if improvement interfered too much with delivery, then the Product Owner had the authority to change an item choice of a team, to balance the goals of improvement and short-term shipping. .

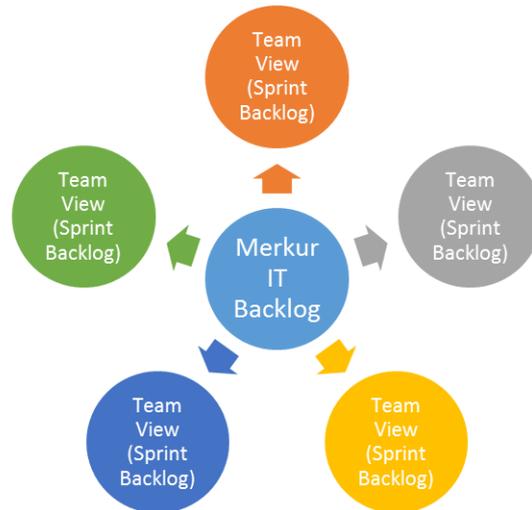


Image 6 - Product Backlog and Sprint Backlogs

One unwanted but necessary side-effect was the tool used for the Product Backlog had to be adjusted. Unfortunately, this turned out to be more time-consuming than other people-oriented activities. It became clear again, that most digital tools are only helpful to a certain degree, but hindering to an even greater extend. So one decision made was that only items larger than simple tasks will be tracked and documented digitally. Tasks were recorded only with simple physical visual management (cards, post-its).

The more important decision was to maintain only one Product Backlog for all domains, applications, and services. For alignment in the larger organization which was still based on a project paradigm, and for legal and reporting reasons, projects were kept as containers for certain endeavours. Therefore a feature-to-project mapping had to be maintained. To keep bureaucracy low, this additional information was added to Backlog Items, and reports added to the tool.

As only one Product Backlog exists for Merkur IT, the tool was configured to support team related views for the Sprint Backlogs. Teams could then concentrate during a Sprint on their selected items for implementation, and for refinement activities they were able to switch back to the overall view.

For the Product Owner and other stakeholders, the overall view was the most important one. Potentially for stakeholders and users also other filter mechanisms could be turned on, e.g. filter per their relevant domain, but this was just for usability purposes to support stakeholders and users.

For strategic decisions this kind of visualizations turned out to be mission critical. However, with all digital tooling peculiarities it was decided to keep analogue representations of it on boards.

6.10 How – Sorting/Prioritization

The Product Owner was and is the sole authority to set priorities. He uses input from teams, stakeholders, users, and other sources for ordering and prioritisation.

The Product Owner was responsible for sorting and adding or removing requirements from the Product Backlog in cooperation with stakeholders, and team-members. Clarification was and is done by the Feature Teams directly with stakeholders. Two of these stakeholders already worked with an existing Scrum team, hence they were able to skillfully continue to add items directly to the Product Backlog.

All other stakeholders did not have authority over the Feature Teams. Prior to LeSS they sometimes had temporary authority over developers, e.g. for the period of certain projects. With LeSS this situation was eliminated.

A risk laying in the larger organization was that an escalation instance, namely the “Gatekeeper Gremium” existed. At a first glance it seemed to be counterproductive to keep this group, which could be called in case of interest conflicts amongst stakeholders. But additional analysis brought to daylight to involve this board of directors in the adoption, and to share information with them was a chance. As a result, this group was instituted as an emergency instance, and as a platform for information sharing. However, further simplification was certainly possible.

This Gremium turned out to be also helpful from a different perspective. The PO was definitely an entrepreneur, innovator, and systems thinker within Merkur IT. In case of conflicting requirements from a group perspective, which were not resolvable by direct interaction, a Merkur group authority needed to be called. This was then the so-called “Gatekeeper Gremium”. With LeSS the interaction with this Gremium changed, as mentioned earlier, but if LeSS would be rolled out to the entire group, some more simplification would be possible.

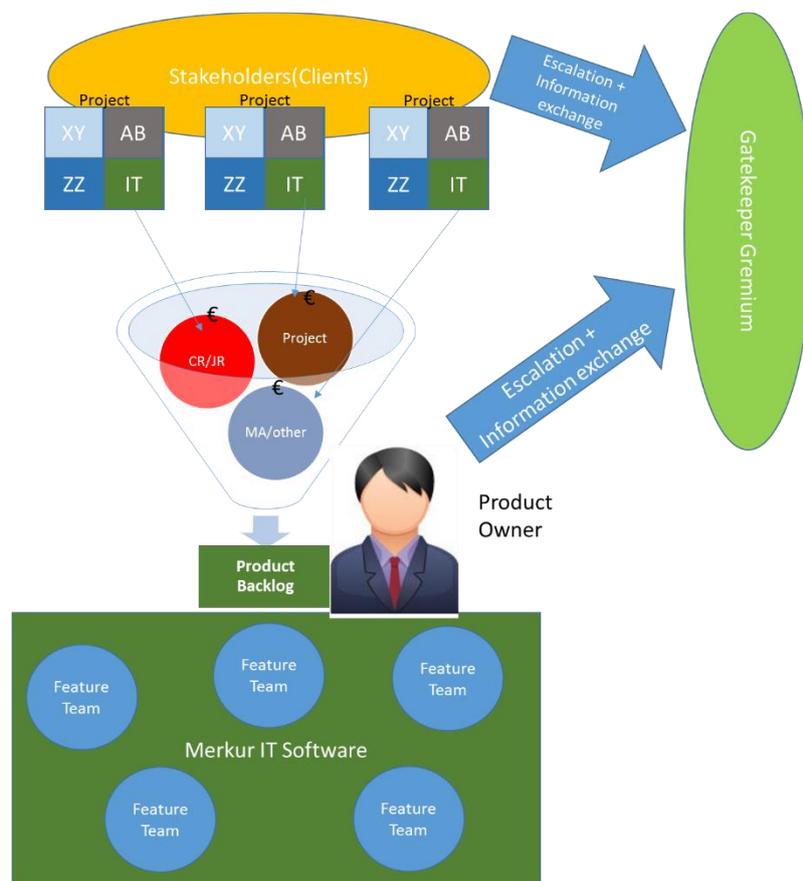


Image 7 - Gatekeeper Gremium - Escalation Instance, and Platform for Information Sharing

Main challenges / key moments:

- Why not having Requirement Areas? This question came up more than once.
- Gatekeeper Gremium of Merkur group – positive or negative to the LeSS adoption?
- PO interaction with stakeholders.

- Elimination of direct control of end-users/stakeholders over developers.

6.11 What/How – What is Product Ownership

As said, one Product Owner was, and is responsible for the entire Product Backlog. Clarification was done by the teams, supported by other knowledgeable people in the organization. Line Management – in contrast to what it comprised before LeSS - became dedicated on simplifying the organization, providing an effective, and healthy work environment, and helping others to get better.

Working with requirements as one aspect of Product Ownership was depicted as followed:

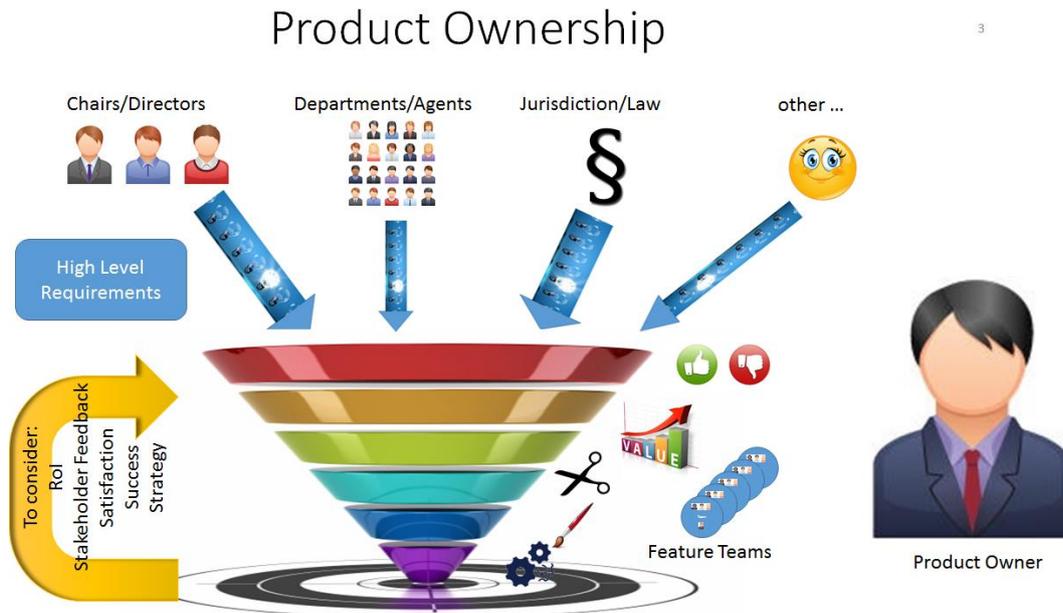


Image 8 - Instances, and considerations in the context of Product Ownership

It was very important for people to understand, that product ownership is a shared activity. The Product Owner should remain the sole person responsible, but to be able to fulfil all required tasks, many skills and hands are required.

Product Ownership was also decided to be based on certain aspects already summarized by Schwaber and Pichler, such as:

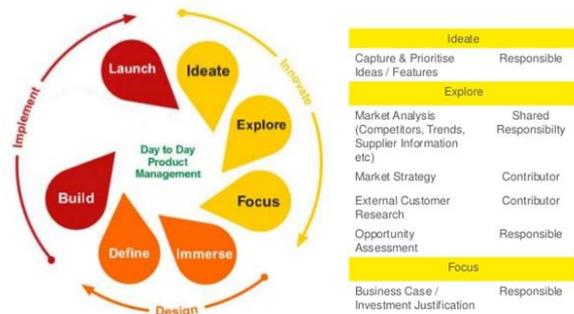
- The Product Owner is the one and only person responsible for managing the Product Backlog and ensuring the value of the work the team performs. This person maintains the Product Backlog and ensures that it is visible to everyone.
- The Product Owner's focus is on return on investment (ROI).
- Understands and shows product management and entrepreneurship capabilities.
- **No Solo Act but Teamwork**

Area	Responsibility
People 	<ul style="list-style-type: none"> • Understand the users and customers with their needs • Collaborate with the development team • Manage the stakeholders
Strategy 	<ul style="list-style-type: none"> • Create the business model • Develop the product roadmap • Describe the user experience and product features
Learning and delivery 	<ul style="list-style-type: none"> • Decide what to do next: Set the sprint goal, detail stories • Collect and analyse data to validate the goal • Update the UX/feature description and business model • Track the project progress (time and budget) • Coordinate the product launch

Image 9 - Product Ownership Overview



Product Manager Responsibilities:



Ideate	
Capture & Prioritise Ideas / Features	Responsible
Explore	
Market Analysis (Competitors, Trends, Supplier Information etc)	Shared Responsibility
Market Strategy	Contributor
External Customer Research	Contributor
Opportunity Assessment	Responsible
Focus	
Business Case / Investment Justification	Responsible

Image 11 - Product Ownership in the context of Product Management

Especially by having a new definition of “product” for Merkur IT, the internal marketing of it changed. It was very important for all people involved to understand, how to communicate and market the new definition. As a consequence, the presentation of Merkur IT and its service to other parties got much attention within the transition.

The PO now became also a “sales-person” for the products. He had to think not only about delivering the right thing, and to think about business value, but also about how to sell what Merkur IT does. In traditional terms often called “project marketing” this set of activities became part of product ownership. Loosely after an important saying in German-speaking countries: “Tue Gutes und Sprich darüber – Do good things and talk about it.”

6.12 How – Approval Process

As part of the concept-to-cash map, the approval process is an existing process, and directed from the wider organization. Prior to LeSS it was a very rigid, step-by-step process, to get budget for projects approved, and assigned to Project Managers. Technicians then were assigned directly to Project Managers, and the Project Managers were allowed to directly instruct technicians. This often led to controversial priority situations, or delays due to unavailability of key personnel for certain endeavours.

With LeSS this process was adjusted. The Product Owner collected all this information from stakeholders and used it for sorting of all Product Backlog items. He, and only he was responsible, that all endeavours were consolidated, rated against each other, and optimized in terms of sorting it, so that Merkur as a whole benefited from it. Also budgets were managed by the PO at Merkur IT. This helped to eliminate the risk of local optimization and the loss or ineffective use of investments to significant degree.

The approval process itself changed from a rigid step-by-step procedure to a checklist for certain points that must be followed, due to legal constraints, compliance prerequisites, and traceability obligations dictated by law. The role of the “Project Manager” changed to the role of the client inside the organization.

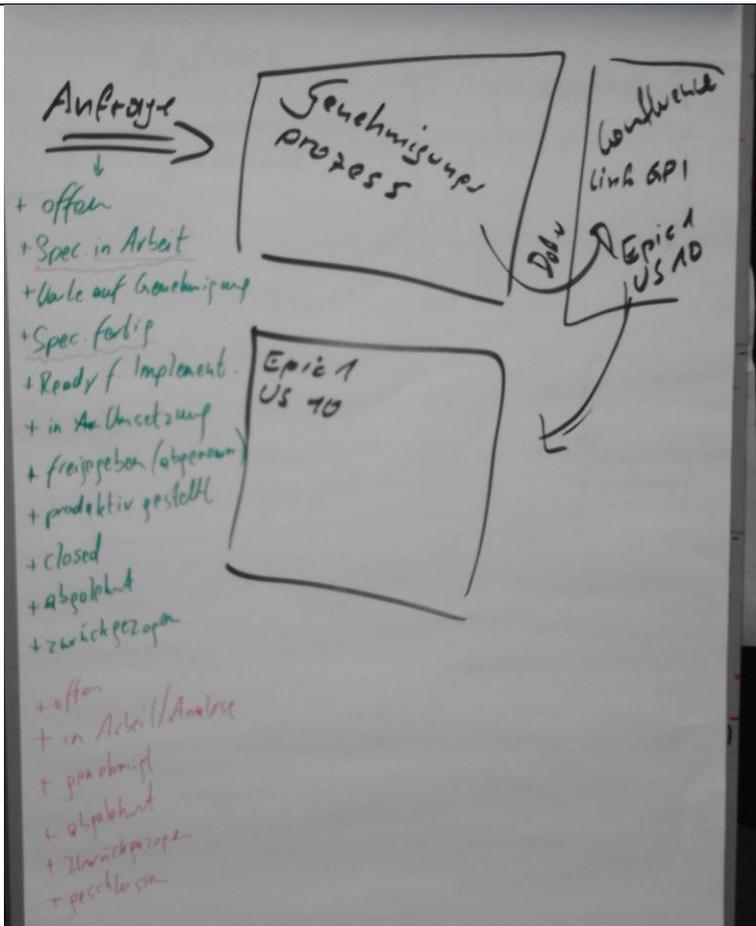


Image 12 - What has to pass the approval process

From request, to concept, to estimation, to order. The approval process changed from a rigid step-by-step process, to a more adaptive checklist.

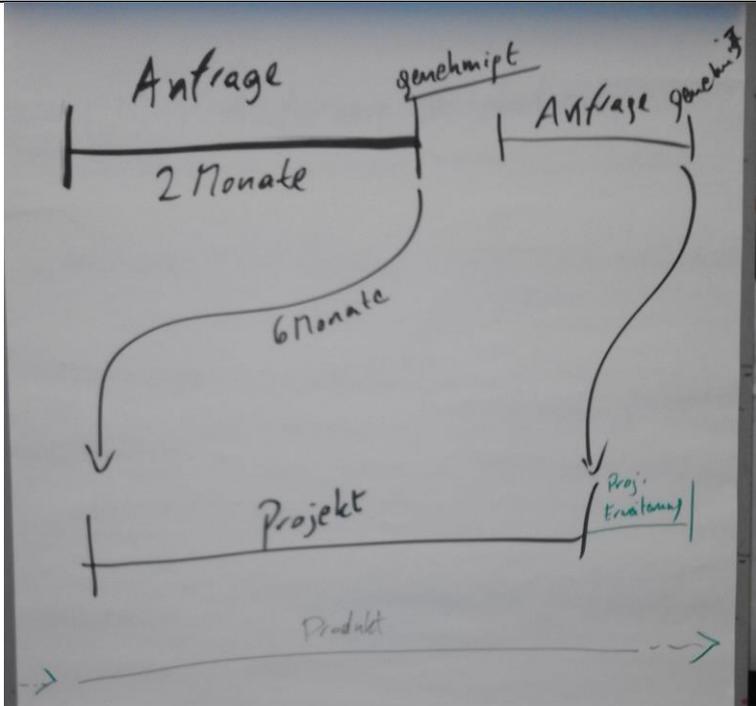


Image 13 - Approval process in context of time-lines, project-lifecycle, and product-lifecycle

Sometime the duration from the first request, to the actual order, or start, can be delayed. Hence the new way of not doing big-design-up front helped to save time and money for Merkur IT, and the Merkur group.

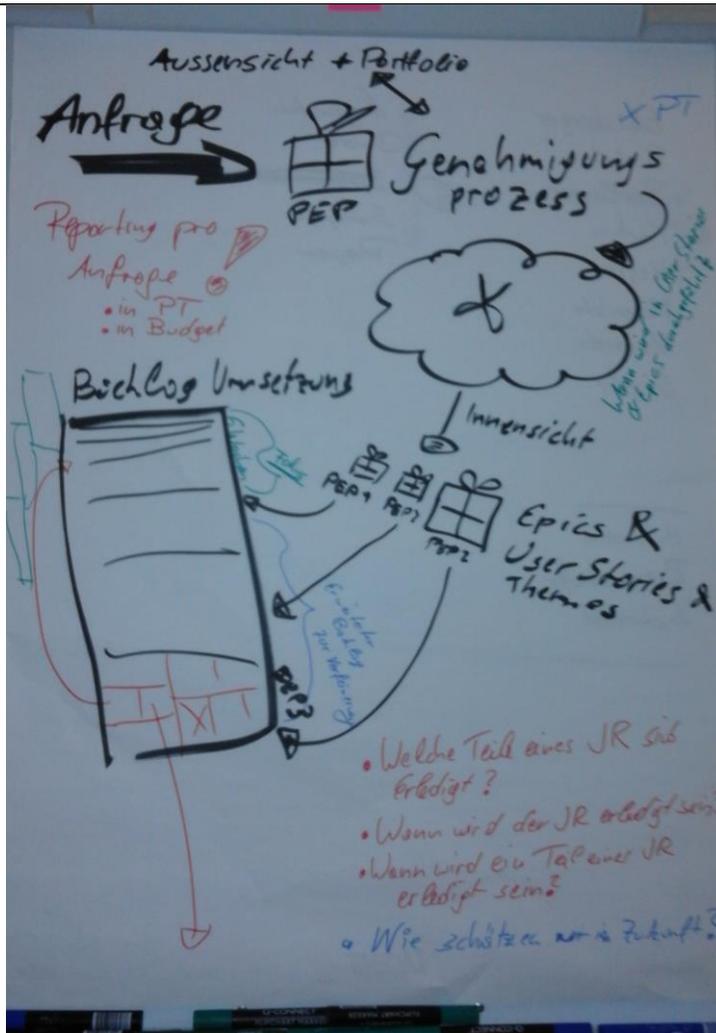


Image 14 - Re-thinking the approval procedure

A depiction of the most important artifacts, and processes to summarize the concept-to-cash lifecycle.

On this flipchart relevant people of Merkur IT simplified a previously rather challenging endeavour to the bare minimum steps required, in order to get an order through. There was still a process-diagram existent, but it became obvious, that it was insufficient, or too strict for various different situations. Hence, in a workshop a simplification was explored, and designed.

Main challenges / key moments:

- The approval process changed to an approval checklist.
- The role of Project Managers changed to the role of clients.
- The overview over all planned, or wanted endeavours improved significantly.

6.13 How – Reporting

Reporting can be a painful obligation. As Merkur IT switched towards agile practices, such as working with story points and velocity, the existing reporting procedures, and indicators had to be adjusted. Before LeSS some KPIs (key performance indicators) were basically micro-management oriented. After LeSS, a switch to more macro performance indicators was achieved.

Unfortunately, it was not possible to entirely move away from more traditional views on progress control, since controlling in the wider organization is very far off agile practices, but it was still a more simplistic, and definitely realistic view achieved by using e.g. velocity together with context-

information for reporting. Since velocity by itself could hold the same risks as traditional KPIs, such as teams start to work more for compliance, or for higher numbers, it was important to put information (e.g. Sprint results) in context to what happened. Hence a Sprint with potentially a very low velocity could still mean good progress, and be reported to controlling as success, if the context shows, that valuable work was done. For many this kind of looking at progress was a huge paradigm shift, and, hence was a topic for further improvements.

6.14 How - Sprints

Sprints at Merkur IT started on Thursday, ran for two weeks, and ended on Wednesday. Bi-weekly meetings (events) held rhythmically were:

- Sprint Planning 1
- Sprint Planning 2
- Daily Stand-ups
- Scrum of Scrums
- Sprint Reviews
- Sprint Retrospectives

Other meetings were held on demand, such as refinement meetings, clarification meetings, sometimes for the departments, and info-meetings for the wider organization. Also events for communities started to arise.

All meetings were visible for everybody via visual management:



Image 15 - All Events

6.14.1 Sprint Planning 1

Sprint Planning 1 was held on Thursday morning. For the first 30 minutes this event was mainly for the PO and the Feature Teams. After that doors are opened for stakeholders/clients.

- Thursday morning
 - Sprint Planning 1: Timebox 9:30 am – 11:30 am

- Intro by PO (explaining Sprint Goal(s)) – 10 Minutes
- Item selected by Feature Teams (representatives) – 5 Minutes
- Info exchange between teams, negotiations, key questions to PO, per team overview – 5 Minutes
- Teams present briefly their key items of their Sprint Backlogs, their highlights for the Sprint, and who across team boundaries they need to interact with, or require help from – 10 Minutes
- Re-arrangement of items in Sprint Backlogs to a customer centric view of domains or customer group related topics (5 Min.)
 - After that stakeholder/clients join
- PO and Feature Teams (representatives) present Sprint Goals, highlights, and interaction requirements to stakeholders/clients – 5 minutes
- Dialogue between team representatives and clients - 30 to 45 min.

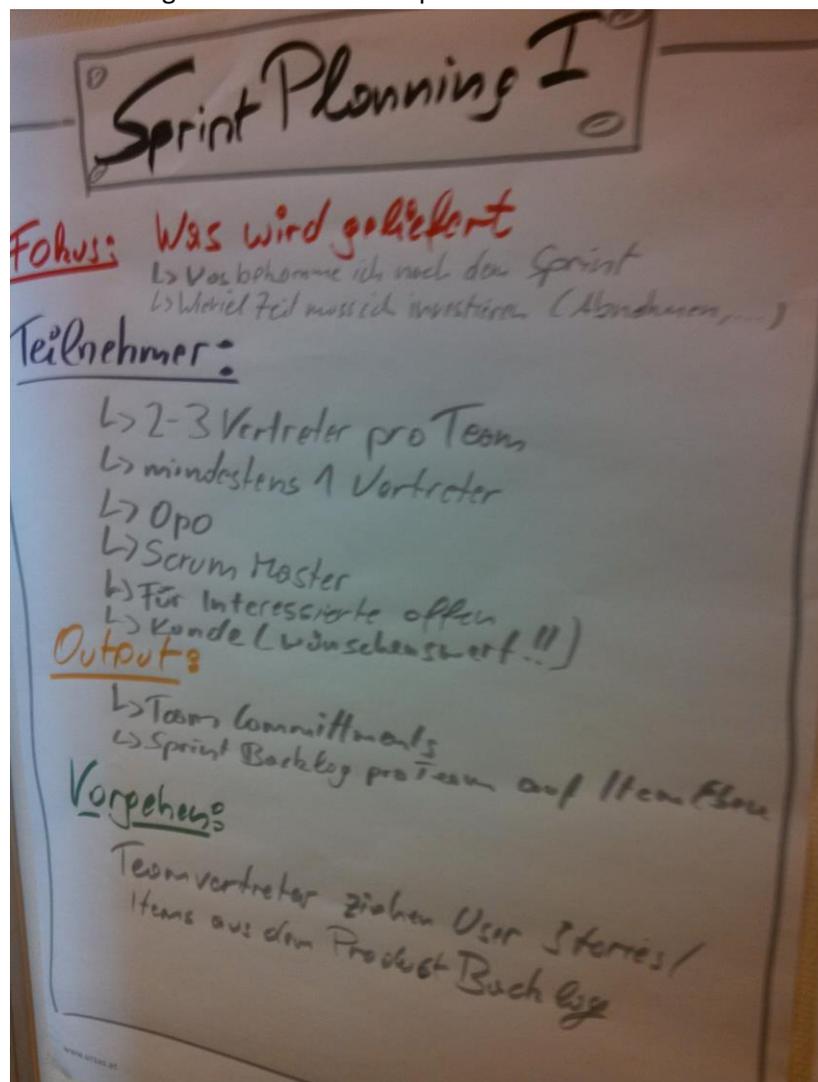


Image 16 - Sprint Planning 1

6.14.2 Sprint Planning 2

Sprint Planning 2 was mainly done by the teams themselves. For a better understanding, and learning purposes, the main event was followed by a consolidation event. This second part was considered to be dropped later, when teams were routined in working together across team boundaries.

- Thursday afternoon
 - Sprint Planning 2: Timebox 1:00 pm – 3:00 pm
 - Note: If a team wants to start earlier, no problem
 - Detailed design/planning, tasks definition by teams in their team-rooms – 90 minutes
 - Tasks consolidation, interaction identification on task level across all teams (representatives) – 30 minutes

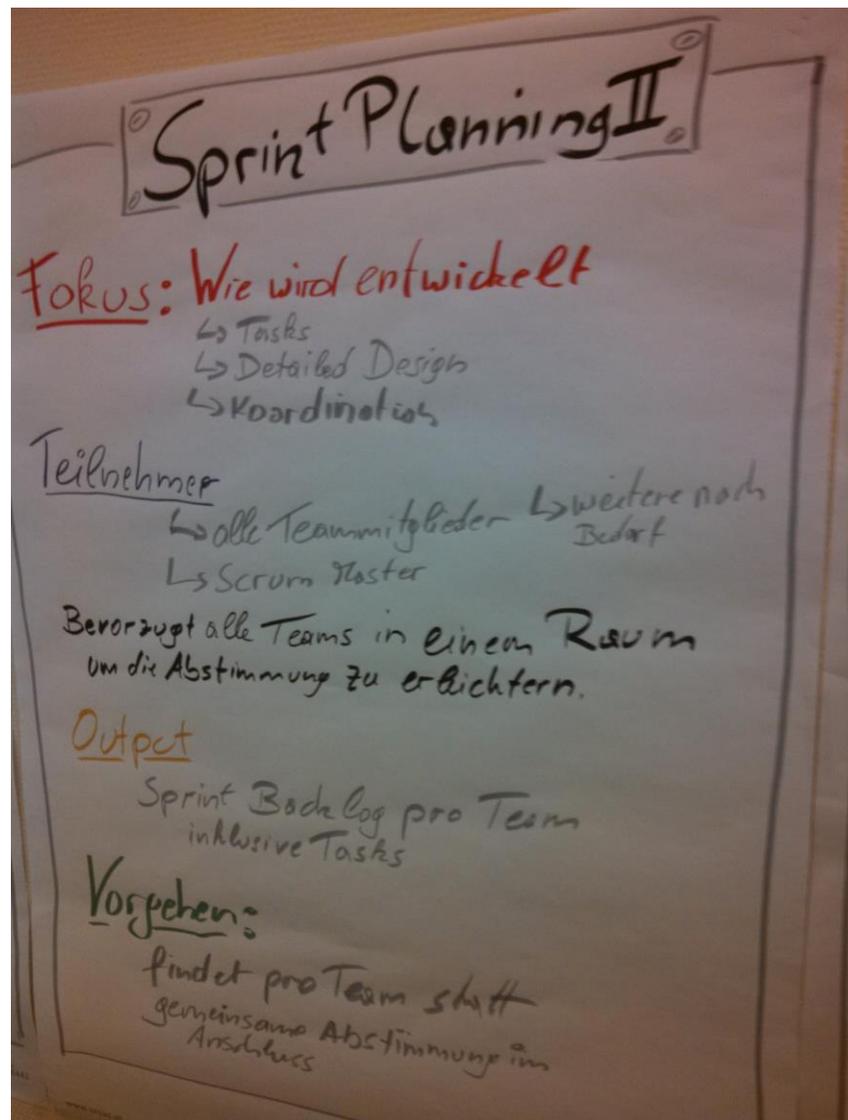


Image 17 - Sprint Planning 2

6.14.3 Daily Scrum

Daily Scrums were held by the teams themselves. Often the Daily Scrums required some rounds to get accepted by team members. Especially to realize the value of it sometimes was challenging for them. Some teams realized the value of this event from the first occurrence on. Especially these meetings showed different group dynamics of the five teams strongly. Also being on-time, and sticking to the time-box without continuing the meeting with other topics was a challenge for some teams in the beginning.

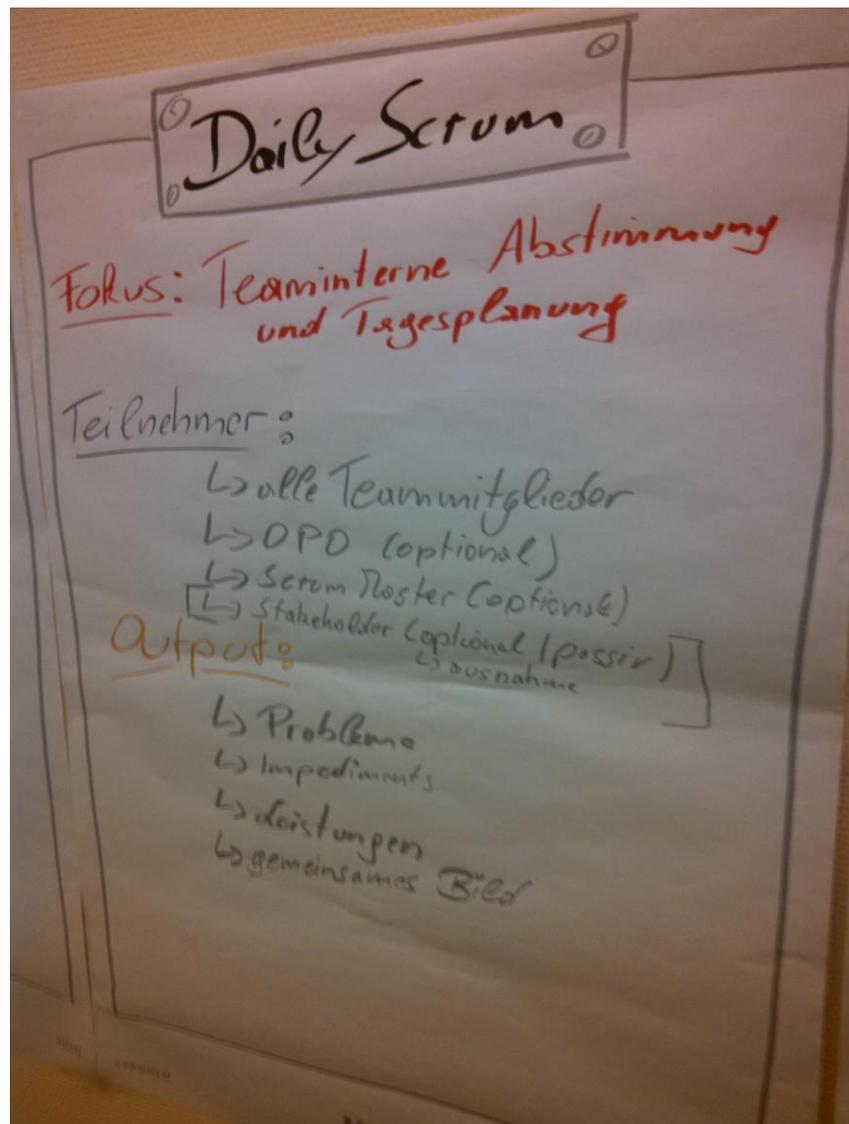


Image 18 - Daily Stand-up / Daily Scrum

6.14.4 Coordination

The Scrum of Scrums was introduced, as the collaboration, and identification of interdependencies across teams was not very mature in the beginning. This event served, as a supportive tool for the teams. It may be dropped later, but even after a while, it still felt like adding value.

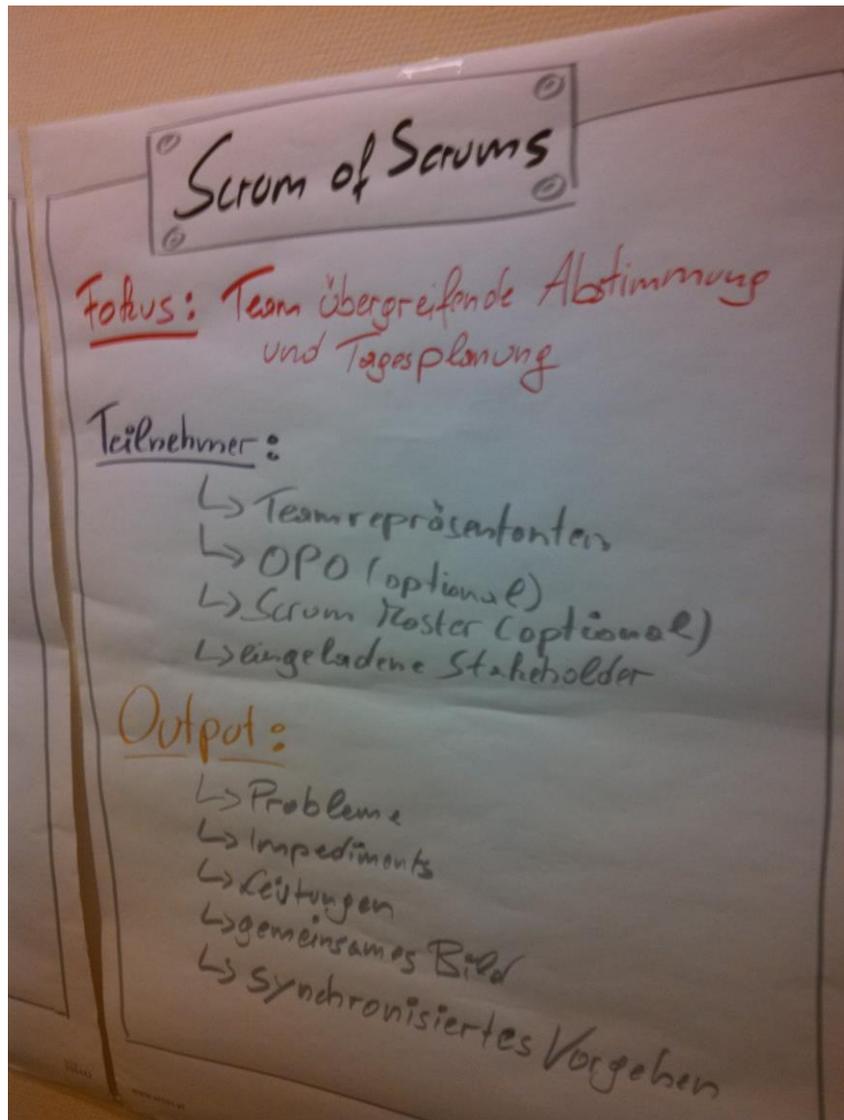


Image 19 - Scrum of Scrums

As coordination between teams within the Sprint, on tasks, design, code integration, etc. is one of the most important aspects of scaling, it was important for all involved to ensure it works. In the beginning Scrum Masters, the PO, and line managers supported the teams in this regard by guiding them towards conversations, asking important questions about cross-relations of features, or sometimes simply calling a meeting to clarify certain details. In this phase it was very important not to walk into the trap of command-and-control behaviours. Self management, team responsibility, and learning was still in the foreground.

What teams did was that they e.g. put tasks about interdependent features on to the task-boards of all teams involved. This helped to identify cooperation requirements for individuals across teams.

Another mechanism to support coordination was to “forget” about team borders in certain Sprints and do Sprint Planning 2 together, refinement together, or code together. Also testing in certain cases was done together to ensure, not just the isolated feature requests were satisfied team internally, but end-to-end processes.

Design boards, User Story Mapping, and documentation also were used to support the product view and foster coordination across teams. E.g. boards very often were located in particular team rooms, but all design sessions on these boards were done by people of different teams.

Sprint Planning 1 and Sprint Reviews were done together with stakeholders, and clients. This required teams to see requirements from a different angle - from the customer perspective. Hence, in Sprint Planning 1 meetings boards showing features requests (backlog items) were grouped by customer domains, not by teams. They held the same information as the Sprint Backlogs, but were grouped differently, so customers could understand what to expect and support in the Sprint.

Representatives of different teams talked together with stakeholders and clients per customer domain. This started coordination for a particular Sprint right away. During the Sprint clarification meetings often were held per customer domains. Hence team members had to think beyond their team borders, and second hand information often resulting in information loss was largely avoided.

In Sprint Reviews the customer domains again drove the presentations of finished features, and allowed team representatives as well as the PO to talk about upcoming Sprints, issues, and other important aspects to the clients and stakeholders.

6.14.5 Sprint Review

The Sprint Review was seen as a mixture of fair of achievements, and a opportunity to talk about upcoming endeavours, issues, and other important points. The focus was still on the Sprint result. It was principally open for everybody, but with the unwritten right for the PO to only invite, who will not create damage. For the first three Sprints participants were only the PO, the Feature Teams (all members), the Scrum Masters, some line-managers, and two stakeholders. It was limited to these roles, as the head of development, the line-manager in charge, wanted to get an impression, whether it helps to transport information across Merkur IT, or just shows chaos.

The first two Reviews obviously left room for improvement. Hence line-management together with the PO and the Feature Teams decided to sort out the main issues first, before opening doors for a wider group. The main issues found were mainly that the teams were not fully aligned, yet. The paradigm shift towards one Product Backlog, and one Product Increment just started, hence people thought too much in silos, still. However, this behaviour started from the third Sprint on to change gradually. People started to request open doors, and more transparency for users, and stakeholders.

The Sprint Review itself was divided into two parts. The first part was designed to provide an overview for a large audience, mainly through the PO and representatives of Feature Teams. The second part was designed to give interested stakeholders, users, and also e.g. prospects, as well as other team members the chance to dig deeper into feature details. As the first part takes place in a large room, the second part was done in the team rooms. Clients, prospects, etc. can go there and try features, ask team members questions about details, or exchange valuable information of any other kind. This second part served already as input for upcoming Sprints, and of course for refinements. It also helped to get to know each other better, to understand each other better, and, hence, to improve the quality and the hit ratio of expectations. As for Merkur IT, and also the wider organization this meeting already showed to be potentially the most valuable improvement in terms of collaboration, and communication between technicians, clients, and users, this event got a lot of attention.

- Wednesday morning
 - Sprint Review: Timebox 9:30 am – 11:30 am
 - Intro by PO (short overview, brief outlook to next Sprint and Releases) – 10 Minutes
 - Overview by Feature Teams (representatives, some details, where to see what) – 10 Minutes

- High-level Q&A – 10 Minutes
 - Switch to team-rooms
- Showing the real stuff in team-rooms, stakeholders, prospects, team-members can walk around and try results out, ask detailed questions, interact with team-members (or simply praise them 📄)

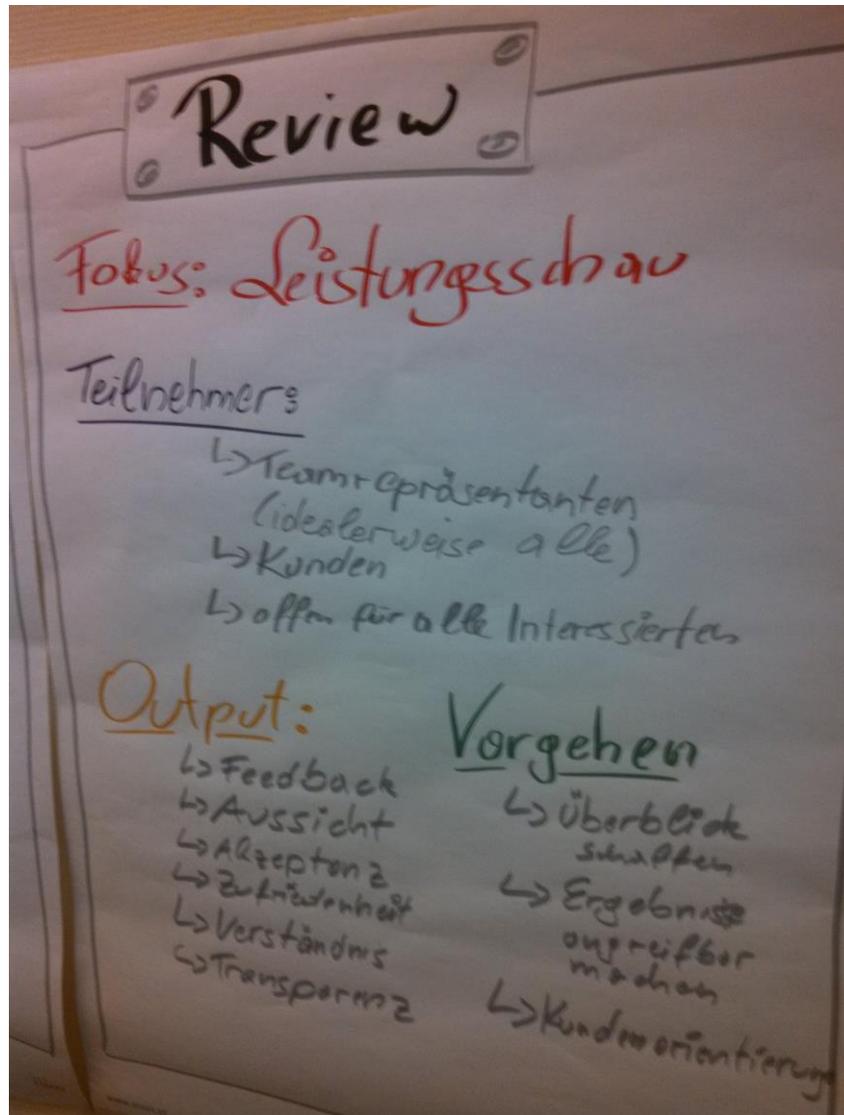


Image 20 - Sprint Review

6.14.6 Sprint Retrospective

The Sprint Retrospective was also split into two parts. The first part was for single teams, or sometimes for two or three teams. The second part was an Overall Retrospective for all teams (representatives) together. Time-wise it deviated slightly from LeSS recommendations, but had still all elements of the two different retrospectives addressed in LeSS. As in Merkur IT there were only two Scrum Masters for five teams, the arrangements were somewhat impeded by this limitation.

Especially the first retrospectives in the first Sprint in LeSS left lots of room for improvement. The Scrum Masters decided to focus on positive aspects of the first Sprint. Team members however expected to use the retrospective for what went not so well. It was a clash of expectations, which was resolved in the second retrospective already. The Scrum Masters sometimes decided to focus on single teams, which seemed to require more assistance, and let other teams perform retrospectives themselves. This experiment seemed to work out rather well.

- Wednesday afternoon
 - Sprint Retrospective: Timebox 1:00 pm – 3:00 pm
 - First part for single teams, or sometimes two or three teams – 90 minutes
 - Overall retrospective for all teams (representatives)– 30 Minutes

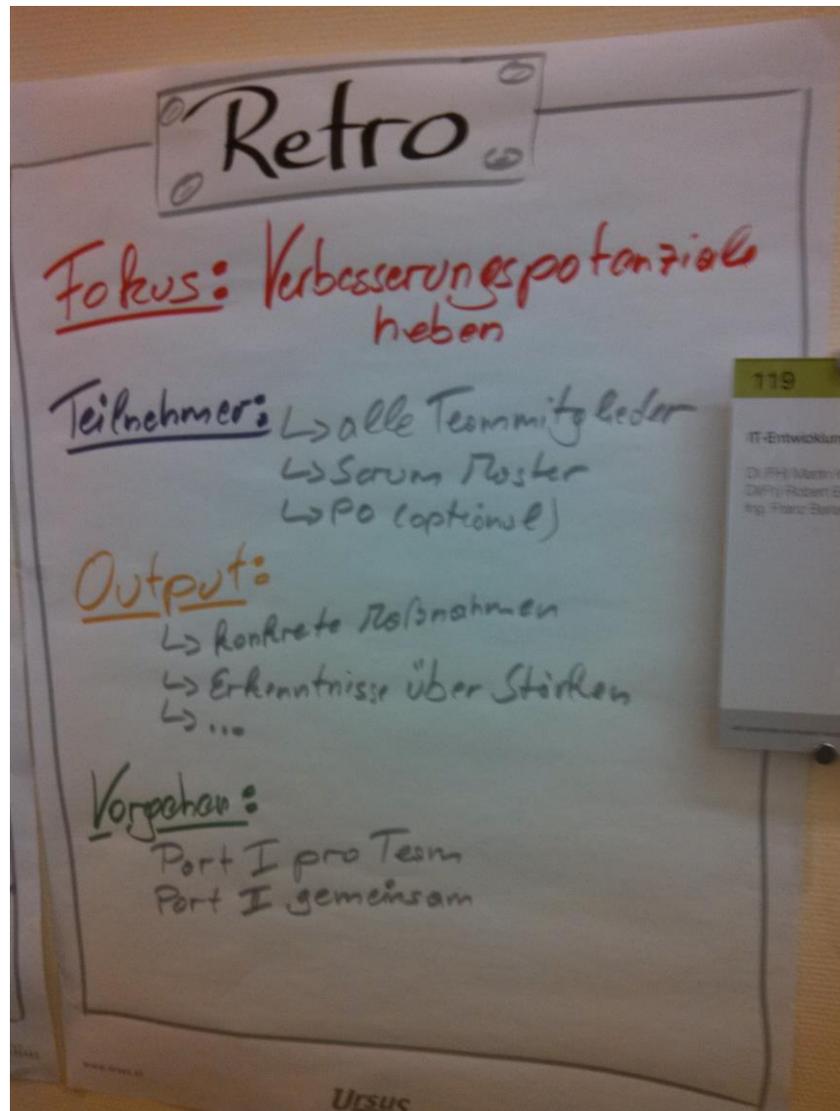


Image 21 - Sprint Retrospective

7 Development Practices / Tools

7.1 Definition of Done

In the beginning the Definition of Done (DoD) was maybe not the most elaborated one. The first Scrum team worked for over two years with a DoD, which became the basis for the other teams. A constant inspect-and-adapt cycle was installed for the DoD, as already people complained about incompatible points for legacy software, or too much room for interpretations if working with different parts of the system landscape. It became clear, improvements were required.

What was interesting to watch was the reaction of people to the DoD. Almost everybody started immediately after sharing the baseline to think about what it means, what other implications to working-style, and techniques it could have. One team replied with an urgency to improve documentation. Also communication-in-code was implicitly mentioned as a potential mechanism for improved information sharing. It seemed from the outside people waited for something like a DoD.

7.2 Test Driven Development

The first Scrum team already worked with TDD (Test-Driven Development) and BDD (Behaviour-Driven Development). Hence, the other teams took on these practices as well. BDD provided scenario level prioritization, and the communication/translation between business people and technicians improved. As a result, the first team, and its previous Product Owner, decided to keep this practice.

As the application (“system”) landscape was very heterogeneous, each Increment required to make changes to various different systems. As this is the component view, from a customer perspective, the Product Increments can be seen as services that stakeholders and users can leverage right away, at least on staging environments. All of them were supposed to be ready for production. However, it must be said that still some level of technical debt existed, even for new features. As we are talking about legacy systems, sometimes with a lots of technical debt, new features inherited that sometimes. However, due to the new way of working, many of these legacy debts in legacy systems have been identified, corrected, and eliminated. With further evolution of the workbenches, and even more modern development techniques the load of technical debt will decrease further. The benefits over the first couple of months of the adoption were already intriguing.

8 Communities

Line-management supported the idea of setting up communities. Ideas were created, some events happened, but at time of writing of this case-study this was still work in progress.

9 Summary

For about two years a single Scrum team existed in a still very traditional plan-driven environment at Merkur IT. The initial thought about working with an agile approach was inspired by the attempted migration of an existing system, that failed more than once.

The agile approach was deemed a last chance. Management made the decision in favour of agile development but started with some reluctance. With the help of JIPP.IT and Dr. Wolfgang Richter as a coach, it turned out to be a very effective endeavour, improved communication with stakeholders significantly, led to fast results, and to unexpected positive outcomes.

After the migration was finished, and some changes in positions in the line-hierarchy of Merkur IT, and in the board of directors of the Merkur group, line-management decided to dare the next step, and to roll-out agility for all development teams. Again JIPP.IT was asked to support this endeavour. LeSS was the choice of framework. Workshops together with line-management resolved un-clarities and the concept for the next steps became tangible, visible and plannable. The latter was very important for the larger organization.

However, the transition was planned like an agile product development endeavour, with inspect-and-adapt as one of the main principles. Line-management very fast accepted the depicted approach, made the decision to go forward, got the board of directors of the Merkur group on board, and started with their exciting and promising journey. Agility, and especially LeSS became the philosophy for all ongoing and upcoming challenges.

Selection of LeSS concepts (experiments) we were able to use:

- Product Owner (main instance for prioritization, and more)
- Feature Teams (cross-functional, cross-component)
- Meetings (Overall Retrospective slightly adjusted)
- Line Management as supporting functions
- Overall Product Backlog
- Flat organization
- Scrum Masters for max. 2 to 3 teams
- Volunteering (to a certain degree)
- Team rooms (as far as the facilities supported it)
- Travellers

Some of the compromises we had to live with:

- Project Managers in the Merkur group (acting mainly as stakeholders and/or customers, some clarifying items and adding items to the Product Backlog)
- Gatekeeper Gremium for more global implications
- IT Operations not fully integrated in the Feature Teams, yet

These list is just a small selection of all the experiments, and changes we were able to achieve, but they were the most influential to the organization, and the adoption at this stage.

It was an interesting journey, and will continue. Operations was not part of Feature Teams, yet, but intended to be integrated. Additionally, other more traditional processes and structures, predefined by the larger organization, were not changeable, yet. However, they were also subject to be inspected and adapted in the long run.

The benefits of LeSS as a (de-)scaling framework with focus on simplification could be seen throughout the organizations. Collaboration improved, communication improved, planning improved, quality improved, and much more. It was fascinating to see what LeSS can trigger.

As a coach I deem this as a very interesting, joyful journey, with many challenges, many positive outcomes. Of course change is challenging for the organization and individuals, but from my point of view it was worth the effort. This is underpinned by the feedback of the majority in the organization, even beyond the borders of Merkur IT. Me, and Merkur IT call our joined endeavour a success story.