2. The roles in Scrum

The term "Scrum" comes from rugby. It refers to the scrum as the *motion of the entire team as a unit* to push the ball towards the opposition's goal. As mentioned above, agile relies on the approach of small, self-organizing teams. Scrum assumes the entire team is responsible for a milestone - meaning a Scrum is a movement of the entire team towards a goal.

Scrum defines three main roles, within which tasks and responsibilities are divided and defined exactly: *The Product Owner, the implementation team* (usually referred to only as *The Team)* and *Scrum Master*. All three together are called the *Scrum team*.

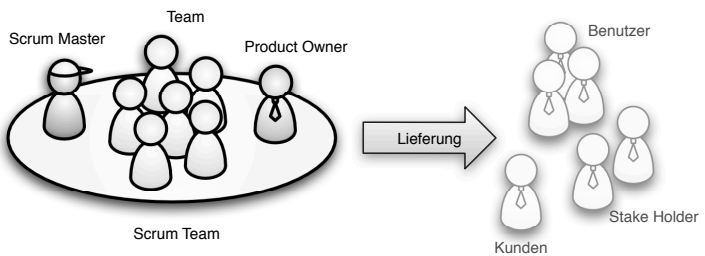

Figure 2.1: The Scrum Team

The collective responsibility of the Scrum team is to deliver a product to customers or stakeholders. Although other people are involved and are important (Managers, other departments, etc...), they are not assigned a role in Scrum and as such, do not have a defined responsibility within the Scrum framework.

The persons allocated to the three Scrum roles should be ideally allocated 100% and not share or play any other role. The Scrum Product Owner represents the business interests (shown with a tie) and the Scrum Master (shown with a cap) can be regarded as a team coach.

2.1. The Product Owner

The Product Owner (P.O.) is responsible for the commercial success of the product to be developed and is therefore responsible for the Return on Investment (ROI) of the development project (or product). He or she ensures the right product is developed and defined, the product characteristics and requirements are refined to specific needs, and ensures the requirements are prioritised according to business needs.

Main responsibility: profitability of the product

The Product Owner works with the customers, users and stakeholders. He or she consolidates the needs of the various stakeholders, and ultimately determines the scope and sequence of the delivered functionality in order to always deliver the best value. The requirements are captured in a list, called the *Product Backlog*, and ranked depending on their business value and the associated technical risks (i.e. potential revenue or cost savings through software).

Because of this responsibility the Product Owner in Scrum is the only person who determines the order of the desired delivery, in other words, the order of the product backlog – even though many factors and people influence it.

Establish a vision

The Product Owner is also responsible for establishing a product vision which can be communicated. The aim is to show a rough goal: what will the product be and why it is being developed. The vision provides guidance to the Scrum team throughout the development of the product. The vision and the Product Backlog capture the ("What?") and not the ("How?").

A significant difference from the classical way of working is the amount of inter-personal communication between the person making the request (the Product Owner) and the people implementing the request (the team). Written requirements (documents, wiki pages, etc.) are allowed; however Scrum calls for direct (face-to-face) communication between team, product owner or customer. As a rule, Scrum is an artefact reduced approach to software development, meaning less documents and more communication. The amount of documentation is at the discretion of the Product Owner or should be determined by the organizational governance.

Accept or reject results

The Product Owner is also tasked to either accept or reject what the team has implemented and delivered. He or she does this by giving feedback to the delivered results - so the implementation team is able to keep themselves informed about the requirements and their collective understanding. Formally speaking, the Product Owner is responsible for the deliveries, but in practice he or she will not decide alone but will also include other parties[1]. The importance of this feedback loop should not be underestimated. Only with this feedback will the team members build extensive domain knowledge, understand more about the products they develop, and learn more about the behaviour of the users.

Agile Requirements Engineering - Backlog Maintenance

The ongoing maintenance of the product backlog is called *Backlog Grooming*, and is another responsibility of the Product Owner. The product backlog is a list that is always changing - it will have new entries added; some will be removed and the order will change. The agile principles state that we want to respond to such changes in order to realise the value of new findings. Since we do not analyze all the details of the requirements in advance, the product backlog is regularly reviewed and refined by the Scrum team. Subsequently, the product owner has a vital interest in cost estimates of individual entries in the backlog, in order to make planning decisions.

In essence, the Product Backlog is constantly in motion and thus requires constant *Backlog Grooming* from the Scrum team.

Release Management

In addition to the active management of requirements, it is the responsibility of the Product Owner to schedule the deliveries of the product. He or she determines at which time the functionalities are delivered. Over several sequential sprints more functionality is developed which can be delivered to the customer. The extent and the timing of the actual delivery is determined by the Product Owner and based on market facts. Although Scrum calls for a

[1] We will investigate this point again in the sprint review meeting

"shippable product increment" at the end of each sprint, this does not necessarily require it to be delivered to the customer[2].

In addition to planning the "Releases" the Product Owner is also responsible for managing the expectations to all parties. After all, he promises a certain level of profitability and determines the order of the requirements in the product backlog. Unrealistic requests or the inability to meet requests from stakeholders in a particular order can, and should be, communicated early in the development process.

No Project Manager

The traditional role of project manager does not exist in Scrum. Even though the product owner takes on many duties and responsibilities, he is not responsible for many of the traditional project management tasks; they are shared with the team and the Scrum Master.

The Product Owner not only has extensive domain knowledge, but also technical knowledge and the ability to make decisions in the context of the project development. He or she must be able to understand the consequences of their decisions.

It is not desirable, but also not uncommon, for a line manager of the team to take on the role of Product Owner. In such a case it is necessary for the line manager to respect the team's self-organization and ensure that personal or departmental objectives do not conflict with the goals of the Scrum team.

2.2. The implementation team

While the role of the Product Owner *mainly* represents the domain knowledge, the technical side is *mainly* represented by the implementation team, usually known only as the "team", and focuses more the "how".

Main responsibilities: translating requirements into software

"*Mainly*" means the team is also responsible to familiarise themselves with the product domain knowledge and contribute to the product backlog to a

[2] In the eyes of „lean", software which is sitting in the company, and not delivered to the client, is seen as waste. The cost connected with this can be identified.

certain extent. However, the main responsibility of the team is to implement the requirements into working quality software.

This implementation of the software is executed within a defined, short period of time, called a *sprint* - the typical length of which is two or three weeks. One sprint cannot of course produce a complete product, so we therefore require several sprints to be executed one after the other, each resulting in an *increment*, a small vertical slice of the product functionality. Whilst it is the role of the Product Owner to be responsible for the value of the delivery, the team has the responsibility for the quality of the delivery.

In order to achieve this the team plans the sprint and makes the decision in this *sprint planning* how many of the highest priority items can be delivered; meaning to be completed, tested and delivered bug free. In addition to this the delivery also means the installation of the software in an environment which is similar to production.

The team is thus responsible for the whole process of converting the idea (request) to the delivered software ("to the customer").

Maintain technical skills

The team is collectively responsible for the full implementation of the product increments within the sprints. In order to achieve this they must possess all the required skills (design, technical skills). Not only are all relevant skills needed in the team, they also need to keep up to date with technology developments. This includes not only their technical capabilities (e.g. knowledge of specific frameworks, etc.), but also more general technical skills, such as design patterns and refactoring, plus agile techniques such as test-driven development[3].

Cross-functional

A Scrum implementation team must be assembled to be cross-functional, i.e. in a team we find developers, testers, designers, architects, etc. In the transition from a traditional organisation structured according to functions, to Scrum, these skills can be initially filled by individuals. However, the introduction of Scrum also leads to expanding the capabilities of a single role,

[3] A detailed introduction to this topic can be found in the book "Agile Developer Skills" [4].

such that functions like testing and development are not covered by one person. Ideally within the team there are no specific roles defined.

Self-organization

Within a sprint the team organizes itself and its work to achieve the *Sprint Goal*. There is no external team manager, team lead or lead developer, to tell the individuals what to do or how to do their jobs. The team in Scrum has its own rules (within company guidelines), to best solve the tasks required.

The team is always striving for continuous improvement. It reflects on the delivery after every sprint, and based on their findings, creates and identifies improvements to the process of delivering software.

Full-time and 7 + / - 2 people

Generally all team members (except for system administrators, special experts, external consultants etc...) should be assigned full time to guarantee continuity and avoid distraction through multi-tasking.

Implementation teams in Scrum are small, ideally around seven people, to ensure effective communication within the team. For teams that are larger than the upper limit of nine people, a drastic decrease in the efficiency of communication and software production is often observed. Also, when teams do not sit together, they typically have poorer communication. In Scrum, it is therefore desirable that a team is located as close together as possible, ideally in the same room.

Quality of delivery

The implementation team in Scrum is responsible for the quality of the software[4]. It is therefore essential that the team will take the necessary measures to ensure quality. This means that the team must carry out functional acceptance tests as part of the sprint, which in turn means that testers are essential to the team.

[4] Regardless of the process model, only those people responsible for the quality of the software can create the software. Tests carried out later, or by members outside the team, prove at best, that the quality is good (or not) - the correction of deficiencies must therefore be integrated into the development process.

2.3. The Scrum Master

In Scrum, the "conversion" of requirements into working software ("from Concept to Cash") is reduced to a few weeks. This results in the visibility of existing blockages, bottlenecks and failures to become apparent to the team and the organization. Scrum is like a mirror to the organization and makes the problems of the development process obvious. The Scrum Master's responsibility is to address these problems.

Main responsibilities: Team Coach

The Scrum Master helps the team solve its problems. He or she plays a management role but that does not mean a disciplinary activity, but everything else: the focus is to ensure the team is working efficiently and effectively. In addition is the need to recognise and drive through conflicts that arise in a new team. The Scrum Master also supports the Scrum team in their constant learning and improvement by creating transparency on all team matters and promoting reflection, but not allowing blame or accusations to be made. He or she assumes the role of facilitator, mentor, coach, but also the mediator, helping the team in its development in order to ultimately be highly productive.

Protecting the team

The Scrum Master protects the team from dysfunctional procedures and processes that produce waste. These are, for example, external factors and constant changes to requirements that are already in progress, or the removal of persons[5] to other projects. He or she makes sure that the team has a protected space (and time), within which to work creatively and productively.

The Scrum Master also protects the team from any friction between the Product Owner and the team, and tries to make the cooperation as effective as possible. This may require that the Scrum Master is also required to support the Product Owner and provide coaching in understanding the full scope of their role.

Removes Impediments

[5] Some organizations people are referred to as "resources" – which is the possible origin of the idea that we could share these resources as required.

The Scrum Master does not only lead, but also serves[6] the team by removing impediments that surface, whilst working through the Scrum framework, which hinder the team to effectively implement the proposed requirements into functional software. These impediments can originate both within the team, such as lack of skills or communication problems, or from external parties, adjacent organizational units or other Scrum Teams. Ultimately he or she helps the team become self-organizing.

Leadership without disciplinary power

The Scrum Master has many tasks that are full filled by a Line Manger in traditional organizations, except for some. One of the tasks of the Scrum Master, which is traditionally neglected, is taking care of the wellbeing of team members; however, this *excludes* any line management or disciplinary actions. It is therefore strongly advised that the Line Manger and Scrum Master is *not* the same person.

The Scrum Master has no formal power; but is a coach for the team. The absence of a typical chief role is a necessary condition for team self-organization and helps build the team's confidence in the Scrum Master. Nevertheless, the Scrum Master is a critical success factor in the overall development project: experience has shown that a well-coached, self-organizing team delivers better software with significantly higher productivity.

The Scrum Master is not the project manager, and it is his duty to educate the organisation about their role and the Scrum framework. Having a dual role, in whatever form (Scrum Master and PO, Scrum Master and Team Member) is an obstacle and a disorder of the effectiveness of Scrum.

Changes in the organization

In addition to his work with the team, the Scrum Master helps the organization cope with changes caused by Scrum. He ensures that the rules of Scrum are considered in the organization and explains the Scrum framework to the affected departments. Usually organisational impediments are initially revealed by the individual work of the team, but no later than when the team

[6] The Scrum Master is often mentioned in conjunction with the term "Servant Leadership", which exactly expresses this ambivalence between leading and serving.

interacts with other units such as marketing and sales, as well as purchasing and accounting.

A process of change[7] begins, and is driven by insights and issues faced by the software development teams. The change also requires guidance from management. It is therefore one of the tasks of the Scrum Master to work with management to ensure that the organization implements these changes (change management).

2.4. Other roles

Beyond the three Scrum roles mentioned above, other individuals or roles contribute to the development of software products. However, they are not part of the Scrum framework, which focuses on the delivery of software in short cycles.

Other roles maybe users, customers, managers, and other groups of people whose input does not make a significant contribution - that is why they are not considered in Scrum with their own roles. This distinction, however, should not be an expression of a lack of appreciation; rather, the stakeholders provide the Scrum team important feedback, ideas, needs and often the necessary budget resources.

Project management tasks

In Scrum, there is no project leader or project manager. The responsibilities and tasks of the traditional project management are divided into the Scrum roles.

The Product Owner has more duties and responsibilities on the side of requirements. He takes over the primary communication with the parties involved, but does not replace direct communication with the team. The Project Manager, who had previously been a technical leader and administered technical tasks, often takes the role of the Product Owner.

The Scrum Master is a coach to the team, facilitates and creates an environment in which self-organization is possible, and is a remover of impediments, so the team can work effectively. Project Managers who traditionally cared for efficiencies and problem solving, often like to take the Scrum Master

[7] I use this like the term "endogenous change" - ie changes that are motivated from within.

role. However, a Technical Project Manager, who used to tell the team members what to do, could become members of the implementation team, however, they need to leave behind old patterns of behaviour and learn self-organization.

Manager

The Scrum framework does not define an organizational structure or process for a company, but rather focuses on describing the Scrum team and its responsibilities. Nevertheless Scrum, which requires self-organization, small cross-functional teams and deliverable product increments, as well as a value system, influences how a company operates.

It is especially important for managers to understand how Scrum works and what happens to an organization. Managers are required to address the problems which become visible through Scrum that cannot be dealt with by the Scrum Master. In addition, managers are required to develop an understanding of self-organization and the related necessary changes required in the organization.