



# Scrum Observer Tools

by Marc Bless

The **Scrum Observer Tools** contain several one-page forms to be used as a Scrum Master, Agile Coach, or any other team observing role you are in. There is one page for each of the Scrum events, for each of the Scrum roles, and for each of the Scrum artifacts. Use these pages to inspect the team's behavior and identify working and missing aspects. These structured observations can be used as input in retrospective meetings or as a basis for your coaching approach with the team.

Most pages are structured as follows:

**Setting** - who are the expected participants of the event, the role, or the artifact, and what is the basic setting

**Aim** - what is the general goal of the event, the role, or the artifact, and what kind of outcome is expected

**Tools and Tips** - which actions could help to set up and improve the event, the role, or the artifact

**Things to look for** - what are the things we would like to observe

**Warning signs** - what are the things indicating misconception or misunderstanding of the original intention

Some of the pages contain different aspects.

I'm looking forward to getting your feedback and ideas for improving the **Scrum Observer Tools**. Drop me a line at [marc.bless@agilecoach.de](mailto:marc.bless@agilecoach.de) with your comments and recommendations.

*This document is compliant to the Scrum Guide edition November 2020. Although the Scrum Guide now talks about accountabilities instead of roles, both authors of the guide still do. So don't be dogmatic on terminology if the intention of Scrum is still met.*

# Table of Contents

<b>Sprint Review</b>	<b>3</b>
<b>Sprint Retrospective</b>	<b>4</b>
<b>Sprint Planning</b>	<b>5</b>
<b>Backlog Refinement</b>	<b>6</b>
<b>Daily Scrum</b>	<b>7</b>
<b>Sprint</b>	<b>8</b>
<b>Scrum Team</b>	<b>9</b>
<b>Developers</b>	<b>10</b>
<b>Product Owner</b>	<b>11</b>
<b>Scrum Master</b>	<b>12</b>
<b>Product Backlog</b>	<b>13</b>
<b>Sprint Backlog</b>	<b>14</b>
<b>Increment</b>	<b>15</b>
<b>Definition of Done</b>	<b>16</b>

# Sprint Review

## Setting

- Scrum Team (Developers, Product Owner, Scrum Master)
- Real users
- Key stakeholders
- max. 4 hours for a one-month Sprint

## Aim

- Answer the question: „
- Presentation of Increments („Done“ items) by the Scrum Team
- Get feedback on the status of the product from attendees
- Spreading knowledge to people who care about the Increment
- Give outlook on upcoming Increments and the release planning of the product

## Tools and Tips

- Invite interested people, real users, and stakeholders on a regular basis
- Plan enough time to prepare and setup before the Review

## Things to look for

- Showing a working product (and/or running tests)
- Transparency of the Sprint Backlog
- Participants know which item or increment is being shown at any time during the Review
- Feedback and discussions from users, stakeholders, and the whole Scrum Team
- New items, ideas, and changes are put into the Product Backlog
- Presenting „what else“ has been done by the team during the Sprint
- Happy users seeing their needs fulfilled
- Appreciate the team for achievements
- Collaboration of attendees to make decisions

## Warning signs

- Surprise party for the Product Owner and stakeholders at the end of the Sprint
- Items are accepted in the Review, not during the Sprint
- Losing time at the beginning due to setup technical infrastructure for the Review
- Use of power point presentations instead of showing a working product
- Blaming the Developers for failing / negative feedback
- Showing unfinished work
- Pure demonstration by the Scrum Team
- No team-external attendees

# Sprint Retrospective

## Setting

- Scrum Team (Developers, Product Owner, Scrum Master)
- safe environment; keep internals intern
- max. 3 hours for a one-month Sprint
- last event of the Sprint

## Aim

- Improve the team's process / daily work
- Improve collaboration
- Closing and re-opening the process' feedback loop
- Identify and decide on change items actionable by the team

## Tools and Tips

- Use the 5 steps: (1) set the stage, (2) gather data, (3) generate insights, (4) decide what to do, (5) closing
- Read books and blogs on „agile retrospectives“ to find new activities
- Focus on very few topics / action items
- Use Traffic Lights to inspect DoD, Working Agreements, and other aspects of the team

## Things to look for

- Surfacing of all issues / problems / impediments
- Deep analysis / root cause of technical issues
- Futurespective / solution-focused goals of issues
- Concrete items how and what to change, actionable in the next Sprint by the team
- Check if action items of the last Retrospective have been done (do this in step (2) gather data)
- Find action items to try something new / experiments
- Scrum Master / facilitator encourages all team members to communicate and collaborate (give room for the silent and shy people)
- Find ways to increase product quality by improving work processes
- Adapting the Definition of Done

## Warning signs

- Running the same agenda / activities every time
- Managers / team leaders joining the Retrospective (to know what's going on)
- People saying: „we don't have a problem, everything is fine“
- Fear of saying the truth
- Team members talk about real issues during the break or after the Retrospective
- General results not being actionable or measurable, e.g. „improve communication“
- Team members talk about „the team“ rather than „we/us“

# Sprint Planning

## Setting

- Scrum Team (Developers, Product Owner, Scrum Master)
- optional: other attendees to provide advice
- First meeting in the Sprint, max. 8 hours for a one-month Sprint
- Input: Product Backlog

## Aim

- Why is this Sprint valuable?
- What can be Done this Sprint?
- How will the chosen work get done?
- Sprint Backlog (contains answers to all three questions)

## Tools and Tips

- Sprint calendar with vacations, days off, etc.
- Identify tasks in the architecture / interfaces / modules on the whiteboard
- Team members explain „what“ and „how“ of each item to each other

## Things to look for

- Define a Sprint Goal as an objective set for the Sprint (Why)
- Define a Sprint forecast by picking items from the Product Backlog (What)
- Plan how to create an Increments from those items (How)
- Add actionable items from the Sprint Retrospective to the Sprint Backlog
- Explore tasks for each item in the Sprint Backlog
- Items and tasks with pen and paper on physical board (or appropriate virtual board)
- Task size is 1 day (6 hours) at maximum
- Scrum Master encourages the team not to over- and under-commit
- Each opinion is respected and discussed
- Items are pulled by priority from the Product Backlog
- Team capacity / calendar is taken into account
- All team members work on identifying tasks for each Backlog item

## Warning signs

- Product Owner presents the technical solution in the Backlog items
- Product Owner defines the tasks for the Developers
- Assign team members to tasks during planning
- Sprint length changes due to holidays or people being absent
- Product Owner neglects to negotiate items when team complains (dependencies, etc.)
- Items/tasks only understood by single experts
- Product Owner is pushing the team's commitment
- Product Owner is pre-defining the Sprint Backlog in detail
- Items are created ad-hoc or on demand during the Sprint Planning
- Sprint Planning is divided into a Sprint Planning I and a Sprint Planning II meeting

# Backlog Refinement

## Setting

- Developers
- Product Owner
- Input: Product Backlog
- Backlog Refinement is NOT a Scrum event but an ongoing activity during the Sprint

## Aim

- Adding detail, estimates, and order to items in the Product Backlog
- Have a ready Product Backlog to pull from in the Sprint Planning
- Gain knowledge and a shared view on the Product Backlog

## Tools and Tips

- Use Story Maps
- Use Magic Estimation techniques
- Use Story splitting patterns
- Bring in subject matter experts from other teams

## Things to look for

- Use pen and paper
- Product Owner talks about „what“
- Everyone actively participates and contributes
- Constructive criticism
- Team offers alternative solutions
- People question the „in order to“ of Stories
- 3Cs: team and Product Owner use Card and Conversation to gain Confirmation

## Warning signs

- Refining too far ahead (2-3 sprints are enough)
- Product Owner insists „this one is a small/easy item“
- Too much analysis („how“) in the refinement
- Splitting items into waterfalls
- Items not ready for sprinting are accepted to be tackled during the next Sprint Planning

# Daily Scrum

## Setting

- Developers
- Scrum Master as facilitator
- optional: Scrum Master and Product Owner participate as Developers if they are actively working on items in the Sprint Backlog
- any interested people (listening only)
- 15 minutes maximum
- Every day, same time, same location

## Aim

- Team coordinates, synchronises and plans its collaboration for the next 24 hours
- Raise impediments
- Are we on the right track to reach our Sprint Goal?

## Tools and Tips

- Use a talking token (stick/ball) to focus on one team member
- Scrum Master stands behind the person talking to avoid being reported to
- Use a timer to visualise the length of the meeting (kitchen timer/sand clock/time timer)
- Use an alarming sound to indicate the start of the meeting
- Experiment with a Sprint Burndown chart if it supports the team

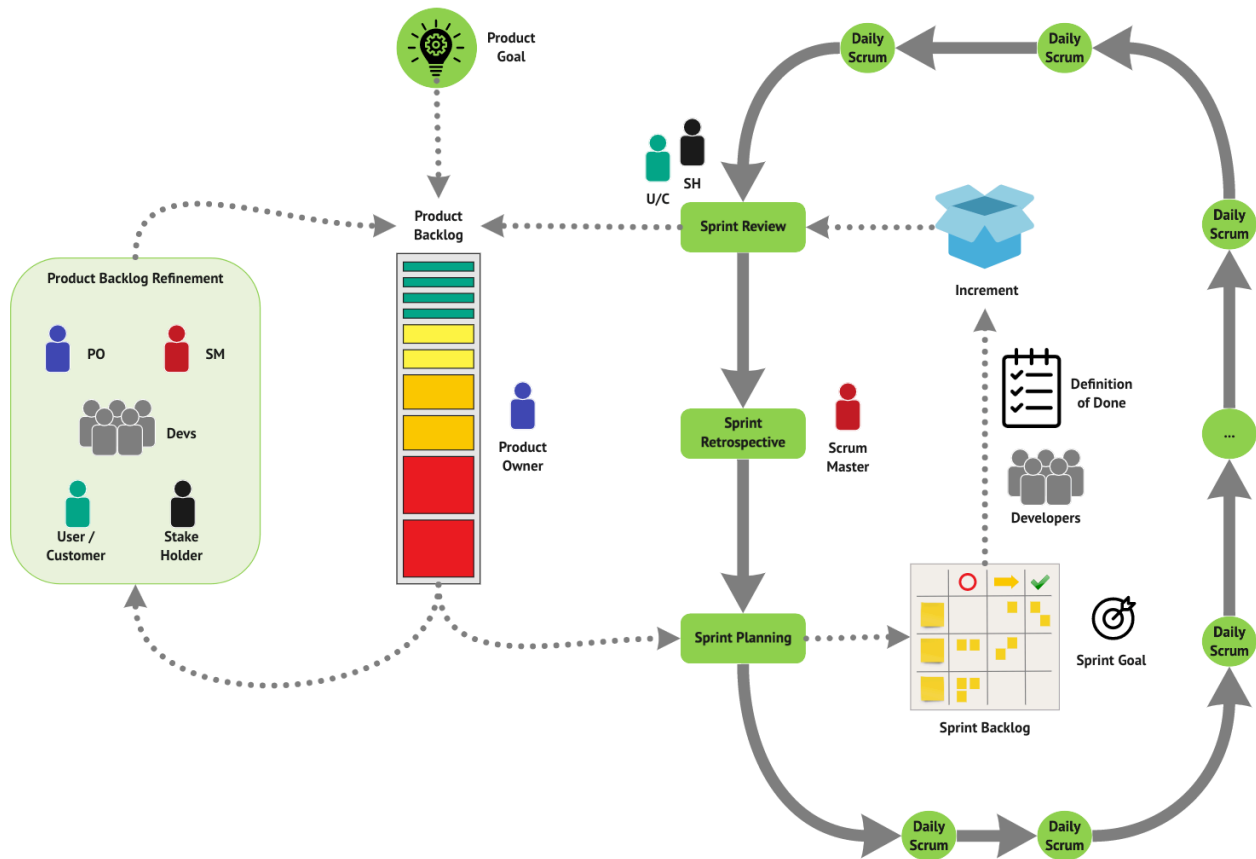
## Things to look for

- Inspect progress toward the Sprint Goal
- Team talks about the Sprint Backlog
- Team works on the board
- People listen to each other
- People work together on tasks
- People do raise impediments
- Daily Scrum starts on time
- Team members meet immediately after the Daily Scrum for detailed discussions, or to replan the rest of the Sprint's work
- Create an actionable plan for the next 24 hours

## Warning signs

- People report to Scrum Master or Product Owner
- Scrum Master or Product Owner delegates/assigns tasks to team members
- Too detailed discussions
- Daily Scrum takes more than 15 minutes
- Time/location changes from meeting to meeting
- Meeting is cancelled due to people missing
- People work on tasks that are not on the board

# Sprint



## Aim

- Focus the team's time to reach the Sprint Goal
- Team creates product Increments
- Repeating time-boxed structure to foster learning

## Things to look for

- Team delivers a product Increment every Sprint
- Team tends to shorten the Sprint over time
- Backlog Refinements takes place as needed

## Warning signs

- Length of the Sprint is adapted according to Sprint scope or team's capacity
- Break between two Sprints
- Changes are made to the Sprint Backlog during the Sprint



# Scrum Team

## Setting

- Developers
- Scrum Master
- Product Owner
- Self-managing team
- preferably 10 or fewer team members

## Aim

- Cross-functional team able to efficiently create effective outcome
- Focused on a Product Goal

## Tools and Tips

- Ask the team!

## Things to look for

- Team members trust each other
- Stable team
- Increments are available at the Sprint Review
- Development Teams are empowered by the organisation to organise and manage their own work
- Split too large teams into multiple Scrum Teams with shared Product Goal, Product Backlog, and Product Owner

## Warning signs

- Team members change on a regular basis
- Other people besides the Developers are needed to create the product Increment
- Sub-teams or hierarchies exist in the Scrum Team
- Teams have unmanaged dependencies to other teams

# Developers

## Setting

- All skills needed to transform the Product Backlog into a valuable product
- Consists of professionals who do the work of delivering Increments of a "Done" product at the end of each Sprint
- Members of the Scrum Team
- preferably 10 or fewer team members

## Aim

- Cross-functional team able to efficiently create effective outcome
- Developers take responsibility for quality

## Tools and Tips

- Ask the team!

## Things to look for

- Team members trust each other
- Stable team
- Developers create a Sprint Backlog in the Sprint Planning
- A "Done" Increment is available at the Sprint Review
- Developers hold each other accountable as professionals
- Developers collaborate on a daily basis

## Warning signs

- Team members change on a regular basis
- Other people besides the Developers are needed to create the product Increment
- Scrum Master, Product Owner, Management or other people tell the Developers how to do their work
- Sub-teams exist within the group Developers
- Team members are deeply specialised
- Team members are not able or willing to support other team members
- Unavailability of team members as reason to defer or delay necessary work items
- Team members have other teams memberships or projects outside the Scrum Team

# Product Owner

## Setting

- Most knowledgeable person for the market and the product to be built

## Aim

- Defining and communicating a Product Goal
- Responsible for maximising the value of the product resulting from work of the Developers
- Responsible for managing the Product Backlog
- Accountable for the outcome of the Scrum Team
- Product Owner takes responsibility for effectiveness

## Tools and Tips

- Let the Developers do or help with managing the Product Backlog

## Things to look for

- Product Backlog items are clearly expressed, ie. all team members have a shared understanding
- Product Backlog items are ordered to best achieve goals and missions
- Developers always work on the most valuable Product Backlog items
- Product Backlog is visible, transparent, and clear to all, and shows what the Scrum Team will work on next
- Developers understand items in the Product Backlog to the level needed
- Product Owner's decisions are visible in the content and ordering of the Product Backlog

## Warning signs

- Product Ownership is spread in a committee
- Decisions of the Product Owner are not respected by the entire organisation
- Other people tell the Developers to work from a different set of requirements

# Scrum Master

## Setting

- Establish Scrum and serve the Scrum Team and the organisation
- Leadership responsibility

## Aim

- Responsible for promoting and supporting Scrum
- Helping everyone understand Scrum theory, practices, rules, and values
- Creating complete transparency of all Scrum artifacts
- Scrum Master takes responsibility for efficiency

## Scrum Master Service to the Product Owner

- Helping find techniques for effective Product Goal definition and Product Backlog management
- Helping the Scrum Team understand the need for clear and concise Product Backlog items
- Helping establish empirical product planning for a complex environment
- Facilitating stakeholder collaboration as requested or needed

## Scrum Master Service to the Scrum Team

- Coaching the team members in self-management and cross-functionality
- Helping the Scrum Team focus on creating high-value Increments that meet the Definition of Done
- Causing the removal of impediments to the Scrum Team's progress
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox

## Scrum Master Service to the Organization

- Leading and coaching the organisation in its Scrum adoption
- Planning and advising Scrum implementations within the organisation
- Helping employees and stakeholders understand and enact an empirical approach for complex work
- Removing barriers between stakeholders and Scrum Teams
- Causing change that increases the productivity of the Scrum Team
- Working with other Scrum Masters to increase the effectiveness of the application of Scrum in the organisation

## Warning signs

- Scrum Master tells the team what to do or how to do its work
- Scrum Master is an additional role of a Developer or the Product Owner

# Product Backlog

## Setting

- Ordered list of what is needed to improve the product
- The Product Backlog is dynamic and evolves over time
- Contains a Product Goal as a future state of the product

## Aim

- Create a shared understanding of the things the team might have to do to develop the product

## Tools and Tips

- Use Story Mapping to visually manage different aspects of the Product Backlog
- Burn Down charts or other tools can be used for Release Planning

## Things to look for

- The Product Backlog changes constantly
- The Product Backlog contains features, functions, requirements, enhancements, and fixes
- Product Backlog items have the attribute of a business value
- Product Backlog items include acceptance criteria to prove their completeness when "Done"
- Feedback from users and customers is incorporated in the Product Backlog frequently
- Backlog Refinement sessions take place with the Developers and the Product Owner
- The top of the Product Backlog contains detailed, smaller items which are ready to be pulled into a Sprint by the Development Team
- The bottom of the Product Backlog contains rough, large items which need refinement and only give a general outlook
- The Product Owner regularly provides information on Release Planning based on the remaining work in the Product Backlog

## Warning signs

- Other sources of requirements are used by the team
- The Product Backlog is defined to be feature complete and gets a „freeze“
- Every item in the Product Backlog is split into small items fitting into one Sprint
- Items pulled into a Sprint Backlog can not be „Done“ at the end of the Sprint
- Every item in the Product Backlog has „priority 1“

# Sprint Backlog

## Setting

- Sprint Goal (why)
- Product Backlog items selected for the Sprint (what)
- Actionable plan for delivering the Increment (how)
- Forecast of the Increment of the Sprint by the Developers

## Aim

- The Sprint Backlog makes visible all the work that the Developers identify as necessary to meet the Sprint Goal

## Tools and Tips

- Use a Burn Down chart to visualise the remaining items of the Sprint Backlog

## Things to look for

- A Sprint Goal is defined
- The Sprint Backlog includes at least one process improvement item identified in the previous Retrospective meeting
- The Sprint Backlog contains enough details/tasks that progress changes are visible in the Daily Scrum
- The Sprint Backlog is emergent during the Sprint
- Unnecessary items are removed from the Sprint Backlog as soon as they are identified
- Developers work collaboratively on the topmost item of the Sprint Backlog

## Warning signs

- Developers work on things that are not represented in the Sprint Backlog
- Scrum Master, Product Owner, Management, or other people outside the Developers change the Sprint Backlog
- Many items of the Sprint Backlog are in progress in parallel (everything is started, nothing gets finished)

# Increment

## Setting

- Increments add value towards the Product Goal

## Aim

- An Increment is a Product Backlog item that meets the Definition of Done
- An Increment supports empiricism at the end of the Sprint

## Things to look for

- An Increment is delivered during a Sprint or latest at the end of a Sprint
- An Increment must be useable and inspectable
- An Increment is a step toward the Product Goal
- An Increment must be in useable condition regardless of whether the Product Owner decides to release it

## Warning signs

- The Developers are not able to create an Increment within a Sprint
- The Developers deliver concepts instead of a working product

# Definition of Done

## Setting

- List of attributes a Product Backlog item needs fulfilled to become an Increment

## Aim

- Shared understanding of what it means for work to be complete

## Tools and Tips

- Start with less attributes and evolve over time

## Things to look for

- Developers get more mature by adding more challenging items to the Definition of Done
- Developers use the Definition of Done on a daily basis to evaluate the progress of the Product Backlog items in the Sprint

## Warning signs

- Definition of Done contains obvious attributes like „Implemented“ or „Compiled“
- Product Backlog items are shown in the Sprint Review although their Definition of Done is not fulfilled