NN/g Nielsen Norman Group

World Leaders in Research-Based User Experience

UX Responsibilities in Scrum Events

Summary: As part of an Agile team, UX professionals should participate in all Scrum events in order to maintain open communication, influence product success, and productively contribute to the team.

By Anna Kaley on April 21, 2019

Topics: Agile, UX Teams, Managing UX Teams

Shifting from traditional product-development processes like Waterfall to modern Agile frameworks such as Scrum can be a challenge for UX. We must learn a whole new set of nomenclature, adapt to new timeframes in which to complete our research or design work, and step outside of our comfort zones to collaborate with cross-functional partners, many of whom we've never worked with before. Once we start making these changes, we quickly realize there's a lot more to Agile than simply working in time-boxed sprints.

Unlike Waterfall, Scrum has many recurring meetings that are typically referred to as events, including **daily standups** (also known as **daily Scrum**), **backlog refinement** (also known as **backlog grooming**), **sprint planning**, **demos**, and **retrospectives**. As UX people move to Agile, they may wonder whether they need to attend each ceremony and what they should do to adequately prepare and participate. In this article, I'll outline what UX should do at each Scrum ceremony to maintain open communication, influence product success, and productively contribute to the team. Much of the discussion in this article will focus on the Scrum framework for Agile, but many of the concepts can be applied to other Agile approaches as well.

TOP

UX Should Be Present and Engaged at Events

UX involvement on an Agile team means more than just working a sprint ahead to quickly and continuously deliver ideas to product owners, stakeholders, and developers. It also means being a present and active contributor in the current sprint, with all its meetings and events. Simply put, UX professionals working in Agile should not skip Scrum events or tune out during them.

When we decide to skip or not fully participate in events, we risk losing sight of:

- The bigger picture, our product vision, and intended outcomes
- Problems we're solving for our users
- A sense of common goals and purpose with our team
- Implementation feasibility and practicality with development
- How our design intent gets translated by developers in code
- What to prioritize in upcoming sprints
- Where discovery or additional sprint 0 work is needed
- Chances to debunk opinion-based design with user-centered evidence
- Opportunities to invite the team to observe research or participate in workshops
- Areas where UX can improve the product
- User stories that need more concrete detail from UX
- Where we may need to support QA with testing

By not attending events, **we also put unnecessary strain on intrateam relationships and communication**. Don't frustrate your teammates by forcing them to inform you of what was discussed, after the ceremony. Many people won't take time out of their busy schedules to catch you up, so engage in the events with a positive attitude to avoid misunderstandings and diminishing relationships.

If you find yourself spread too thin and unable to attend events, you're likely trying to do too much in a given sprint. Reprioritize your workload to favor what needs to get done in the current sprint to ensure that developers have what they need for the next one. Shifting your mindset this way should start to free you up to attend the events. Over time, as you practice balancing time to complete your work with time spent in meetings, you will be able to add more activities back into your sprint workload. Agile is all about responding to change, so **it's important to check in with yourself and make sure you're not overdoing it to the point where you can't make it to events.**

If, after all this, you're still unable to make it to events, discuss your workload with your manager, product owner, and Scrum master to adjust your time accordingly. Your teammates should be your allies and understand why your presence and participation is crucial in these important events. If they don't, educate them on why it's productive for the team and the product to have UX represented as the voice of the user in Scrum events. Without that voice, the team risks introducing UX debt and having to rip out, redesign, or refactor features that were not developed with end users in mind.

What if You're Not Dedicated to a Product Team?

Participating in events is easier when UX supports a single product team, but when it must support multiple development teams, conflicting schedules can make it challenging for UX to attend every ceremony. Try your best to still attend all Scrum events, especially when your projects move from discovery into delivery.

Your presence in events should be active and participatory, as opposed to simply showing up to say you did. Be sure that your attendance adds something valuable to the conversation, whether you bring insights from your research or use your speaking time to **actively check in with developers** who are working on stories that you've designed, to see if they need clarification. If you have conflicting meetings or competing workload priorities, rely on your product owner or Scrum master to keep you informed of what the team discussed in a missed ceremony.

For example, if daily standup on one team happens in the morning, but you have sprint planning at the same time for another team who is in the middle of delivering an important release, favor the sprint planning meeting, but check with

TOP

your product owner or Scrum master after to understand any blockers on the other team that UX can help resolve. In these instances where UX must divide and conquer, it is perfectly acceptable for UX people to rely on teammates to stay as informed and engaged as possible.

Prioritizing Amidst Direct Conflicts and Resource Constraints

When it's impossible to attend all events due to time conflicts or resource constraints, figure out which meetings are not crucial for UX, and occasionally skip them. Ask yourself the following 5 questions to help prioritize direct conflicts with Scrum events and manage the reality of being unable to attend all meetings, for all teams, all the time.

- **Will my absence from the ceremony inhibit or delay the development team in reaching the sprint goal?**
  - If yes, attend the ceremony in question
  - If not, it's acceptable devote your time elsewhere

- **Do I have critical information to share with the team that could significantly affect our product and the experience of end users?**
  - If yes, attend the ceremony in question
  - If not, it's acceptable to devote your time elsewhere

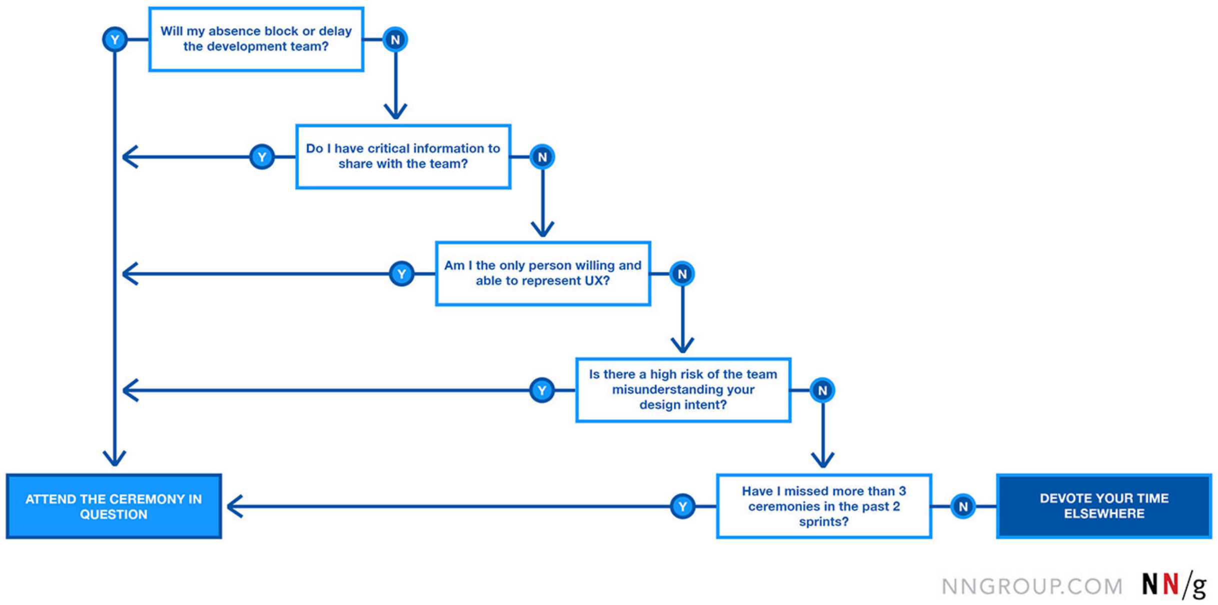
- **Is there another person in the ceremony who can represent UX and update me, afterwards?**
  - If yes, it's acceptable to devote your time elsewhere
  - If not, attend the ceremony in question

- **Have I had enough collaborative time with developers to feel confident that they understand my design intent and evidence from user research?**
  - If yes, it's acceptable to devote your time elsewhere
  - If not, attend the ceremony in question

- **Have I missed more than 3 events in the past 2 sprints?**
  - If yes, attend the ceremony in question

TOP

  - If not, it's acceptable to devote your time elsewhere


Will my absence block or delay the development team?
Y
N
Do I have critical information to share with the team?
Y
N
Am I the only person willing and able to represent UX?
Y
N
Is there a high risk of the team misunderstanding your design intent?
Y
N
Have I missed more than 3 ceremonies in the past 2 sprints?
Y
N
ATTEND THE CEREMONY IN QUESTION
DEVOTE YOUR TIME ELSEWHERE
NNGROUP.COM NN/g

When UX gets spread too thin, use this decision tree to determine if you should attend a Scrum event or skip it in favor of another critical activity.

What UX Should Do in Each Scrum Ceremony

Standup (Daily Scrum)

Most Scrum teams have a daily meeting that's referred to as standup or daily Scrum. (Scrum is both the name of an Agile framework and the name of a daily event on many Agile teams.) The meeting is intended to be very short, usually no more than 15 minutes, and focuses on communicating with the team to ensure that work is on track and any impediments are shared and quickly resolved. **Each member of the team, including UX, remains standing to encourage concise answers to each of the following questions:**

- **What did you do yesterday?**
  - Discuss areas of the design you worked on
  - Outline any collaboration with cross-functional partners that happened
  - Discuss any research you planned, conducted, or data analyzed

TOP

    - Share simple research findings or user insights with the team, when relevant

- **What will you do today?**
    - Keep the team aware of what you plan to do that day related to design, user research, design reviews, workshops, design sprints
    - Mention areas where you’d like input from engineering or product
    - Propose collaborative methods such as structured sketching or whiteboarding sessions
    - Invite team members to usability testing and other research happening that day
    - Hold yourself accountable to getting the work done by announcing, committing, and affirming to your group that that’s what you’re doing

- **What is in your way?**
    - Bring up any internal or external factors blocking you from getting work done
    - Lean on your Scrum master to help you remove these barriers to progress
    - If you need to have specific discussions with team members, suggest a quick turn-around meeting after the standup

Since UX is typically working a sprint or more ahead of the engineering team, it is already a challenge to keep the team aligned. It is for this reason that attending daily standup and clearly and concisely communicating what UX is doing are so important. Additionally, UX should listen to other teammates during standup for any potential blockers or issues that UX may be able to address or help resolve.

Backlog Refinement

Backlog refinement (also sometimes referred to as backlog grooming) usually happens halfway through the current sprint. The goal of this ceremony is to review the items in the backlog to make sure they are ready for the upcoming sprint-planning meeting (and ultimately, the sprint).

TOP

In this meeting, the product owner will review backlog items in their current order of priority and readjust priorities to meet business and user needs. The product owner will lead the discussion with the team to understand if additional stories are needed or if existing ones need to be edited or removed. In some cases, the team will rewrite or add detail to stories together to ensure everything is covered.

Having UX professionals present in this meeting to facilitate whiteboard sketching or other ideation techniques can help the team solve issues together. Spikes may also be added to the backlog for items where the team needs to do additional discovery work in the sprint to get answers to open questions.

When backlog items reach an acceptable level of detail and context, it's important that they also meet your team's definition of *ready to be pulled into a sprint*. This requirement safeguards the team from pulling in sprint tickets that lack the information needed to do productive work, thus having to spend what could have been work time to understand what should be done in the first place. The *definition of ready* for each backlog item will usually vary by team, but should pass the following UX criteria if the outcome is user-facing:

- User story and acceptance criteria are clearly articulated in the ticket
- Visuals and lightweight documentation are available
- Solution has been tested with users and no serious issues emerged
- Design adheres to style guide, pattern library, or design system
- The item has been reviewed by UX, development, and product owner

UX participation in backlog grooming can bring foresight into what the development team will work on in the next sprint and provide additional support for the product owner. Though product owners should be as informed of user needs and business opportunities as UX, they have many responsibilities that they need to juggle (e.g., implementation realities from the engineering team, strategic goals from business stakeholders, core performance metrics of the team), so there may be times when your product owner needs additional help and accountability from UX. UX should help ensure that product owners make the

TOP

right decisions for users and the product when choosing which backlog items come next.

Sprint Planning

Sprint planning usually occurs on the first day of the sprint, or sometimes, one day prior. The goal of sprint planning is to review the prioritized backlog items again, and estimate the level of effort needed to complete them. Much like refinement, the product owner leads this discussion, seeking critical input from the team with every ticket reviewed. Ultimately, the outcome of sprint planning is a sprint backlog that reflects the team's velocity — the amount of work the team is comfortable committing to complete in the sprint. Sprint planning is also when teams set a realistic sprint goal, which is a short, one- or two-sentence description of what the team plans to achieve during the sprint. The sprint goal serves as the team's vision for the next few weeks.

UX involvement in sprint planning is important because, just like development work, UX work should be accounted for in the backlog, either with separate UX tickets that share epics with engineering, or as part of the same tickets where UX work is organized into subtasks off of the parent user story. When estimating your UX work, use t-shirt sizes, rather than story points. if your scrum team uses velocity only for engineering work, which is common. If UX work is included in the team's velocity, participate in estimating story points using the same method as your engineering team.

TOP

S
M
L
Can Do Quickly
No Testing Needed
More Time Needed
Only Item in a Sprint
NNGROUP.COM NN/g

Using t-shirt sizes rather than story points is a good way to estimate the UX work required for an item without impacting the engineering team's velocity. An item of size S requires little UX work, while an item of size L requires a lot (and may actually take up all the UX resources for the duration of a sprint). Tracking UX workload is important to ensure that UX won't take on too much during a sprint, thus introducing a risk of holding up the development team in future sprints.

Regardless of the ticket structure used, UX work needs to be visible to the rest of the development team to ensure that research and design tasks are not skipped. Participating in sprint planning helps to keep the team aligned and informed about what UX will do during the sprint. This approach can help the engineering-team and product-team members recognize the rigor that goes into the design and research process; consistently seeing and hearing about the UX work required for each item will help cement buy-in later on for new approaches and suggestions for changes.

Make sure to account for user testing in sprint planning so that the team knows what to expect and has the opportunity to attend the sessions or debriefs. For larger research initiatives or workshops such as design sprints, where it's important for the [whole team to be involved](https://www.nngroup.com/articles/ux-scrum/), propose committing to a slightly lower velocity that sprint so that engineers have time set aside to attend the

TOP

research without feeling like they must multitask during it to still achieve the sprint goal.

We also recommend that you **treat participant recruiting as an important component of your estimated workload.** Writing screeners and recruiting users can often take a long time, so the more systematically you approach your recruiting, the easier it will be to run studies when you need usability data. Recruiting takes planning and you want to avoid having it be the bottleneck in the process. Maintain a list of potential users, adding to it wherever possible. Try tapping your current customer base, placing ads, working with market-research firms, or including additional questions on forms or surveys to build up your pool of study participants.

At the end of sprint planning, the team should have their workload in order for the sprint. However, remember that the **user stories discussed in sprint planning are intended to be placeholders for further conversation and collaboration with your engineers.** Don't just attach designs to user stories and then throw them over the wall and hope for the best; stay engaged with your team during the current sprint and be available for questions or discussions.

Sprint Review (Demo)

Sprint review (often called demo) usually happens on the second-to-last day of the sprint. In many cases, this ceremony falls on the day before the code is released, if the team is launching at the end of the sprint. During this meeting, the product owner will show working code to stakeholders and others who may be impacted by the work the team accomplished in the sprint. **The sprint review is a good opportunity for UX to support the product owner in presenting the work completed.** It's also a favorable forum to **review test results** and **interject user-centered insights** that either informed the current release or will positively impact the next iteration of the product.

Sometimes, UX may leave the sprint demo feeling overwhelmed by the number of new feature requests or less-than-ideal feedback received. When this happens, document the feedback in an organized way, such as tagging it with categories such as *to-do*, *to-clarify,* and *to-persuade*. Work with your product owner and

TOP

scrum master to help navigate the feedback and prioritize new requirements that arose from demo. It may be that your team needs to have early, frequent meetings with certain cross-functional partners during the sprint to avoid surprises or off-putting expectations.

Retrospective

Post-sprint retrospectives often happen on the last day of the sprint and are intended for the team to review how things went. Ideally, a neutral representative such as a Scrum master from another team, or perhaps a project manager from another part of your organization will run the retrospective to keep the conversation flowing and unbiased. Each team member will discuss what went well in the sprint, what was problematic, and what could be done differently to improve the effectiveness of the Scrum team.

The retrospective is a great time for UX to talk about process changes; because the topic of the meeting is to reflect on the team's processes, people will be more receptive to suggestions. In developer-focused environments, it can be difficult to advocate for process changes that will improve the ability to deliver good UX work, so consider positioning your ideas as experiments to be tested during the next sprint.

Retrospectives are also appropriate times to discuss:

- Situations where you didn't get the background information or research time you needed to design something properly
- Implementation mistakes or miscommunications where the developers didn't implement your designs how you had intended
- Questions about your deliverables and the *definition of done* that you provided for designs
- Ways in which you communicated design specifications: if they were problematic, confusing, or not sufficiently specific
- Having product, engineering, and business stakeholders observe usability-testing sessions

TOP

In addition to sharing why the past sprint could have gone better, listen for ways to improve how you communicate with the rest of the team. Are your design specifications easily understood? Did you overwhelm your team with user data that is irrelevant for the current sprint? Were your design solutions challenging to implement with existing architecture or frameworks? Focus on whether you are helping the rest of the team solve their own particular problems, because **that's how UX can be seen as a real asset, rather than a bottleneck.**

Retrospectives are also a good time to bring up the user's perspective. Oftentimes, teams will have to make tradeoffs where the user experience might be deemphasized in favor of getting work shipped during a sprint, so discuss with your team if any of the recently completed user stories contain UX debt or compromises from a user-experience perspective.

If your team isn't holding retrospectives, then your process is not Agile. It's up to every team to find its own way, but the true value of the method is only realized when the team finds out what works and what doesn't. Because there are so many retrospectives in the life of an Agile team, chances are that eventually some UX-related suggestions will be listened to. **A good rule of thumb is that the team shouldn't implement faster than it can learn.**

Conclusion

It's easy for UX professionals to question whether they should attend Scrum event and, at times, UX may even perceive these meetings as barriers to productivity. However, UX should be involved in Scrum event to stay engaged and aware of what's going on with the team. For more on balancing UX workloads alongside responsibilities in Scrum event, take our full-day Lean UX and Agile course.

Share this article: Twitter | LinkedIn | Email

TOP

Consulting

Expert Review

User Testing

Customized Research

Team Training

Applied Workshops

Keynote Speaking

Free Guidance

Articles & Videos

The NN/g UX Podcast

About

Why NN/g

About Us

People

Contact

Return Policy

TOP

Cookie Preferences / Cookie Declaration / Privacy Policy

TOP