Chapter 22. Sprint Retrospective

Scrum provides two inspect-and-adapt opportunities at the end of each sprint: the sprint review and the sprint retrospective. In the previous chapter I discussed the sprint review, where the team and stakeholders inspect the product itself. Let's now turn our attention to the sprint retrospective, where the Scrum team examines the process used to build that product.

I begin with an overview of the purpose of and participants in the sprint retrospective. I then describe the prework and major activities associated with a sprint retrospective, the most important of which occur after the sprint retrospective when the participants actually follow through on the improvements they identify.

Overview

In the preface to his book *Project Retrospectives*, Norm Kerth, the founder of the modern-day movement on retrospectives, summarizes the purpose of retrospectives by quoting a passage from *Winnie the Pooh* (Kerth 2001):

> *Here is Edward Bear, coming downstairs now, bump, bump, bump, bump, on the back of his head, behind Christopher Robin. It is, as far as he knows, the only way of coming downstairs, but sometimes he feels that there is another way, if only he could stop bumping for a moment and think of it.*

Sprint retrospectives give the whole Scrum team an opportunity to stop bumping along for a moment and think (see Figure 22.1). Inside the timebox of the retrospective, teams are free to examine what's happening, analyze the way they work, identify ways to improve, and make plans to implement these improvements. Anything that affects how the team creates the product is open to scrutiny and discussion, including processes, practices, communication, environment, artifacts, tools, and so on.

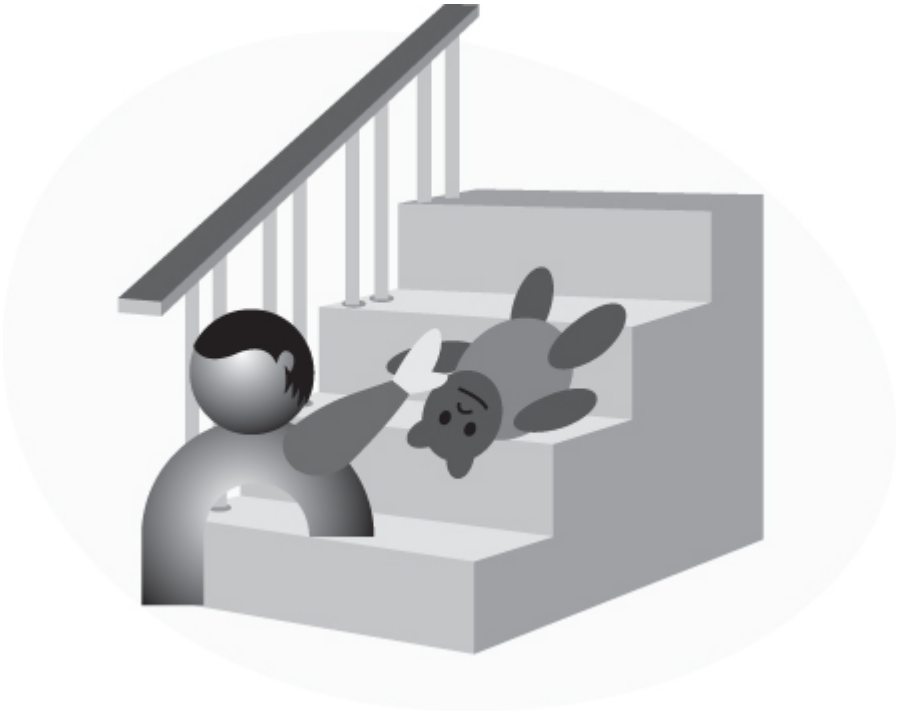

Figure 22.1. Edward Bear illustrating the need for a retrospective

The sprint retrospective is one of the most important and least appreciated practices in the Scrum framework. It is important because it gives teams the chance to customize Scrum to their unique circumstances. It is underappreciated because some people have a misguided view that it takes time away from doing “real” design, build, and test work.

The sprint retrospective is a crucial contributor to the continuous improvement that Scrum offers. And while some organizations might wait to do a retrospective until the end of a large development effort, Scrum teams hold retrospectives each and every sprint (see Figure 22.2), allowing teams to take advantage of insights and data before they are lost.

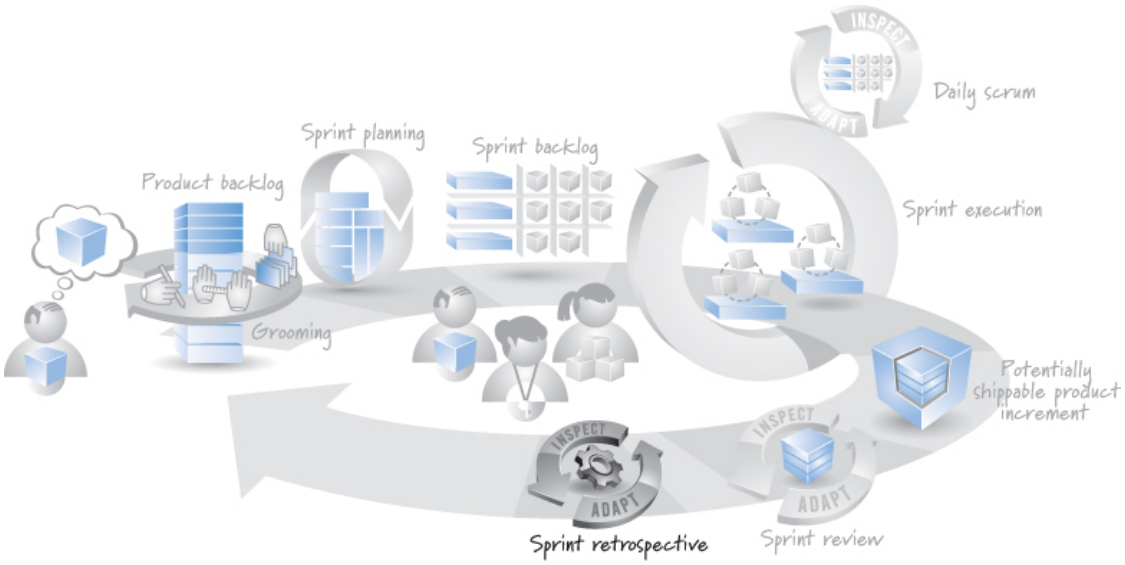

Figure 22.2. When the sprint retrospective happens

Because a Scrum team meets at the end of each sprint to inspect and adapt its Scrum process, it can apply early and incremental learning throughout the development process and thereby significantly affect the outcome of the project.

In the rest of this chapter I describe a detailed approach for performing sprint retrospectives. However, don't let the details mislead you into believing that a sprint retrospective is a heavyweight, ceremonial process. A sprint retrospective can be as simple as the Scrum team members coming together to discuss questions such as

- What worked well this sprint that we want to continue doing?
- What didn't work well this sprint that we should stop doing?
- What should we start doing or improve?

Based on their discussions, team members determine a few actionable changes to make and then get on with the next sprint with an incrementally improved process.

Participants

Because the sprint retrospective is a time to reflect on the process, we need the full Scrum team to attend. This includes all members of the development team, the ScrumMaster, and the product owner. The development team includes everyone who is designing, building, and testing the product. Collectively, these team members have a rich and diverse set of perspectives that are essential for identifying process improvements from multiple points of view.

The ScrumMaster attends, both because she is an integral part of the process and also because she is the process authority for the Scrum team (see Chapter 10). Being an authority doesn't mean that the ScrumMaster should tell the team how to change its process. Instead, it means that she can point out where the team is not adhering to its own agreed-upon process and also be a valuable source of knowledge and ideas for the team.

Some argue that having the product owner at the retrospective might inhibit the team from being completely honest or revealing difficult issues. While this can be a risk in some organizations, the product owner is a critical part of the Scrum process and as such should be part of discussions about that process. If there is a lack of trust between the product owner and the development team, or there is a low level of safety so that speaking candidly isn't comfortable, perhaps the product owner should not attend until the ScrumMaster can help coach those involved toward creating a safer, more trusting environment.

Assuming trust and safety are reasonably in place, an effective product owner is critical to achieving the fast, flexible flow of business value and therefore should participate in the sprint retrospective. For example, the product owner is the channel or conduit through which requirements flow to the team. What if something is wrong with how requirements are flowing through the Scrum process? Perhaps PBIs are not well groomed by the start of sprint planning. In such cases it would be difficult for the Scrum team to brainstorm potential process improvements if the product owner were absent from the retrospective.

Stakeholders or managers who are not on a Scrum team, on the other hand, should attend a retrospective only if invited by the Scrum team. Although transparency is a core Scrum value, the reality is that many organizations have not yet achieved a level of safety to support non-Scrum team members regularly attending retrospectives. The team members must feel safe if they are to have an open and candid discussion without feeling inhibited by outsiders. If the team doesn't feel safe enough to reveal the real issues because outsiders are attending, the retrospective will lose its effectiveness.

Prework

Prior to the sprint retrospective there is some prework to complete (see Figure 22.3).

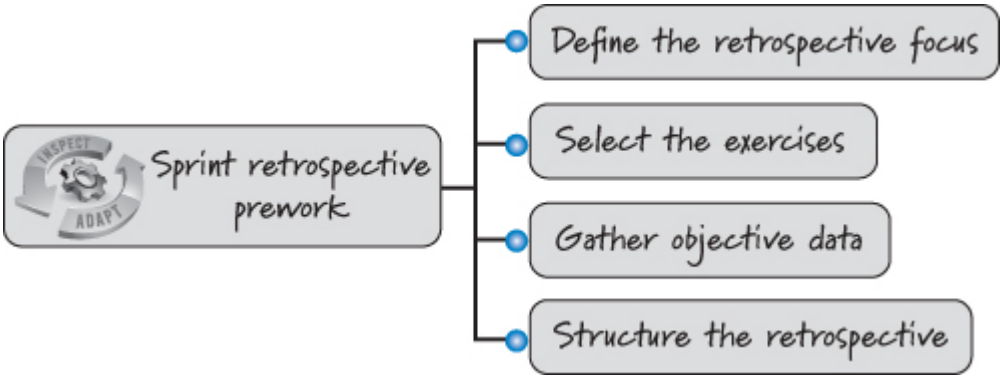

Figure 22.3. Sprint retrospective prework

For short-duration sprints or for teams that are using a well-practiced, simple retrospective format, this prework should not require much, if any, time.

Define the Retrospective Focus

Each sprint retrospective should have a well-defined focus. The default focus is to review all relevant aspects of the process the Scrum team used during the current sprint. However, there are times when a team might select a different retrospective focus based on what is currently important to the team and where it is energetic about seeing improvement. For example:

- Focus on how to improve our skills with test-driven development (TDD).

- Focus on why we build what we think the customers want, but when they see it they frequently believe we misunderstood their desires or missed an important facet of the requirement.

Establishing and communicating the focus before the start of the retrospective allow the Scrum team to determine if any non-Scrum team members should be invited. In addition, knowing the focus before the start of the retrospective allows the team to select appropriate retrospective exercises and gives people time to gather and prepare any data needed to ensure a smooth performance of the retrospective.

Having the ability to define a specific focus can help long-lived, high-performance Scrum teams continue to extract measurable value from sprint retrospectives. For example, in one organization I coached there was a mature Scrum team whose members had been working well together for nearly three years. They had gone through many dozens of sprints together. They were starting to feel that doing a sprint retrospective focused on the just-completed sprint was often low value. One team member remarked, “For a long time the sprint retrospectives felt indispensable, and now they frequently just feel like process for the sake of process.” What we ended up doing was performing shorter, more focused sprint retrospectives that allowed the team and invited outsiders to dig into very specific issues, going deep into root cause analysis. The result was that the team continued to learn and improve despite its considerable experience with Scrum. There is always room for growth; it just might require a more focused retrospective to uncover it.

Select the Exercises

Once we have established the focus and final participants for the upcoming retrospective, we can determine which exercises might help participants to engage, think, explore, and decide together. A typical retrospective includes the following exercises:

- Create and mine a sprint event timeline.
- Brainstorm insights.
- Group and vote on insights.

However, we might choose to vary these exercises to support a particular focus or set of participants. We might also decide to try new exercises to keep things fresh. See *Project Retrospectives* (Kerth 2001) and *Agile Retrospectives* (Derby and Larsen 2006) for additional exercises.

The participants don't have to decide exactly which exercises to use during prework. In fact, it might actually be better to select some exercises in a just-in-time fashion during the retrospective based on what the participants think would work best. At the same time, some exercises, especially those that require data or supplies, are best determined during the prework. Be prepared but stay flexible.

Gather Objective Data

Because a sprint retrospective is performed in a focused, short period of time (many teams establish a timebox), any legwork to collect needed data should be done before the retrospective begins.

We know both the focus and the exercise options for the upcoming retrospective, so we should have a good idea of what, if any, objective data should be gathered. Objective data is hard data (not opinions), such as what events happened and when, or counts of the number of PBIs that were started but not finished, or the feature burnup chart for the sprint illustrating the flow of completed work. At this point we are not organizing or analyzing any data; we are just collecting it so that it is available during the retrospective.

Structure the Retrospective

Like sprint reviews, retrospectives happen at the end of each sprint, often immediately following the sprint review, and generally should recur at the same place, day, and time each sprint. However, unlike for sprint reviews, you might occasionally need to vary the place, date, or time of a particular retrospective to better serve its focus, any non–Scrum team participants, or specific exercises you are planning to run. That's why I like to review the structure of the retrospective as part of the prework.

The exact length of the retrospective is influenced by factors such as how many people are on the team, how new the team is, whether any team members are located remotely, and so on. In my experience, teams new to Scrum have a tendency to budget too little time for their retrospectives. It's difficult to hold a meaningful sprint retrospective in less than 60 minutes. As a rule, I usually budget about 1.5 hours for the sprint retrospective when using two-week sprints, and proportionally more when using longer sprints.

The Scrum team should choose a sprint retrospective location that is most conducive to achieving a successful outcome. Some teams prefer to hold their retrospectives in the standard team area where their big visible charts are located. This gives them easy access to a wealth of relevant information. Others prefer to meet away from the standard team area, perhaps to introduce an environment with less emotional saturation where people might feel less inhibited and more likely to speak freely. Again, location doesn't matter nearly as much as the fact that you are meeting in a safe environment where team members feel free to speak their minds.

Although the ScrumMaster will often act as and can be quite effective as the facilitator for the sprint retrospective, any capable team member can fulfill the role of retrospective facilitator. There are also times when bringing in a skilled, neutral, outside facilitator might be the best solution to help team members either get started doing retrospectives or to assist them through a particularly difficult or sensitive retrospective where a closely aligned, internal facilitator may be far less successful. Alternatively, in organizations with multiple Scrum teams with different

ScrumMasters, it is often helpful and enlightening to all involved to have the ScrumMaster of one Scrum team facilitate the retrospective of a different Scrum team. We should establish who is going to facilitate the retrospective during the prework.

Approach

Figure 22.4 illustrates the sprint retrospective activity.

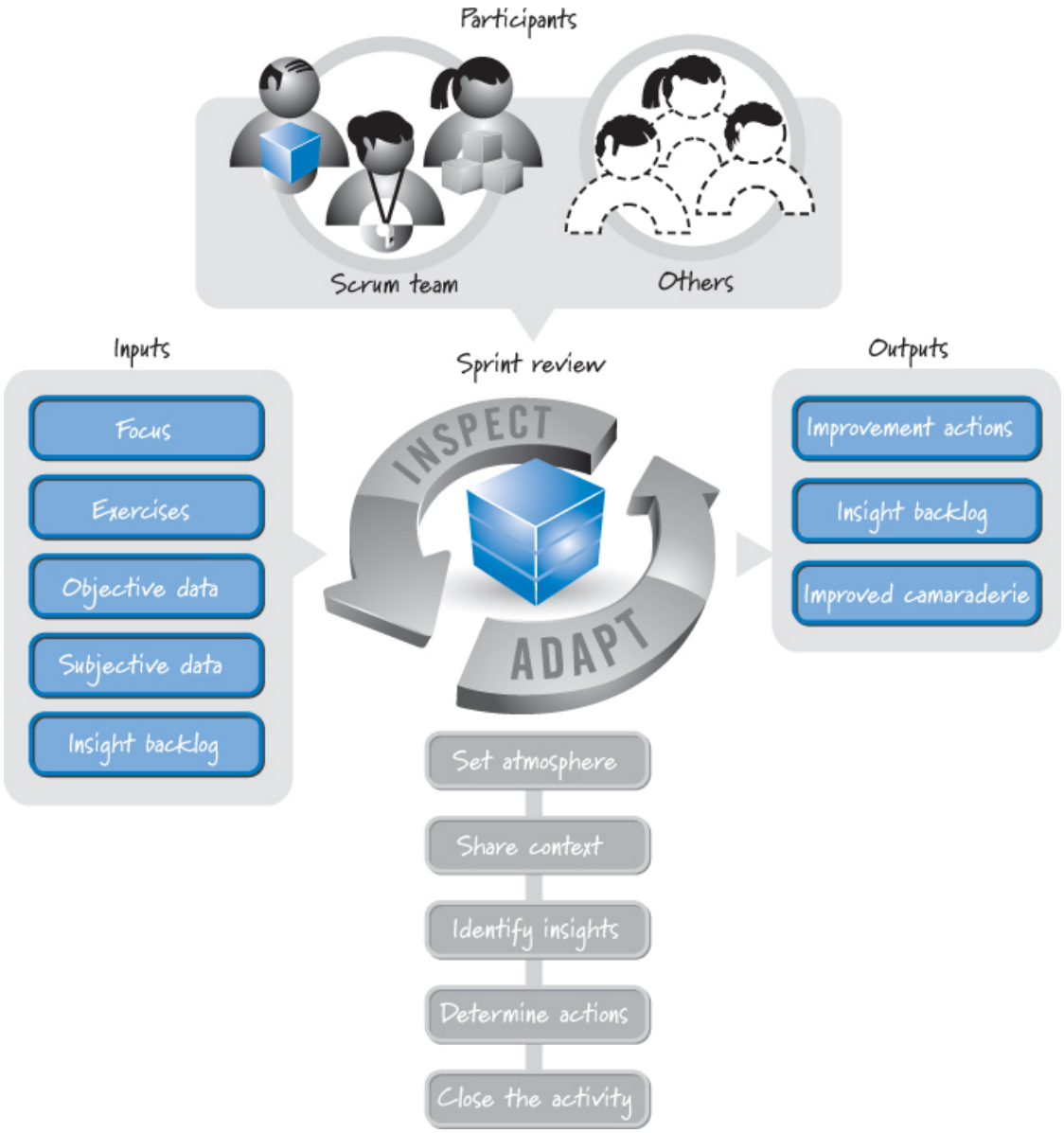

Figure 22.4. Sprint retrospective activity

Inputs to the sprint retrospective include the agreed-upon focus for the retrospective and any exercises and materials that the team might decide to use during the retrospective. In addition, most retrospectives require

at least some precollected, objective data. And one piece of input every attendee will bring without fail is her own subjective data regarding the current sprint. Another retrospective input is a backlog of insights produced in previous retrospectives.

The outputs of the sprint retrospective include a set of concrete improvement actions that the team has agreed to perform in the next sprint. The outputs might also include a backlog of insights collected during the current retrospective that the team will not address in the upcoming sprint but might choose to address in the future. Team members should also expect improved camaraderie as an output from a retrospective.

While many retrospective approaches exist, most seek to answer the following questions:

- What worked well this sprint that we want to continue doing?
- What didn't work well this sprint that we should stop doing?
- What should we start doing or improve?

An approach (similar to one described by Derby and Larsen 2006) that I find useful is to set the atmosphere for the retrospective, create a shared context among the participants, identify insights that can lead to improvements, determine concrete improvement actions to take during the next sprint, and close the retrospective. These steps are shown in Figure 22.4 and explained in the paragraphs that follow.

Set the Atmosphere

During a retrospective, people are being asked to analyze the behavior and performance of their team and to make specific recommendations for how the team can improve itself. Putting the team (and by extension oneself) under a microscope can be an uncomfortable experience. So, a good way to start the retrospective is to establish an atmosphere that makes people feel comfortable participating.

People must feel it is safe to express their opinions without fear of retribution. Teams should have established ground rules, or a working agreement, which make it clear that expressing opinions and airing dirty laundry are safe things to do. It is helpful for the ground rules to make clear that the focus is on the organizational system and process, not the individuals, thus making it safe to explore what went wrong.

There will be times when problems are people problems; the retrospective is not the place to solve them. The retrospective is about improving the Scrum team's process, not about assigning blame or reprimanding individual behavior. When setting the atmosphere, ensure that the ground rules reinforce the concept of a blame-free environment.

It is also important to establish a precedent of active participation. We won't have a very effective retrospective if people assume a passive role. So, when setting the atmosphere, it's a good idea to get people talking just to prime the pump of participation. Some teams do something as simple as ask each participant to express in a few words her current feelings or energy level. It's not critical what question people are asked to answer, but that they are asked to say something to get in the mood of talking.

Share Context

A group of people can all experience the same event and yet interpret it quite differently. To successfully inspect the current sprint, it is important to get everyone on the same page so that they have a shared context.

To establish a shared context the participants must align their diverse individual perspectives (see the left side of Figure 22.5) into a shared team perspective (see the right side of Figure 22.5).

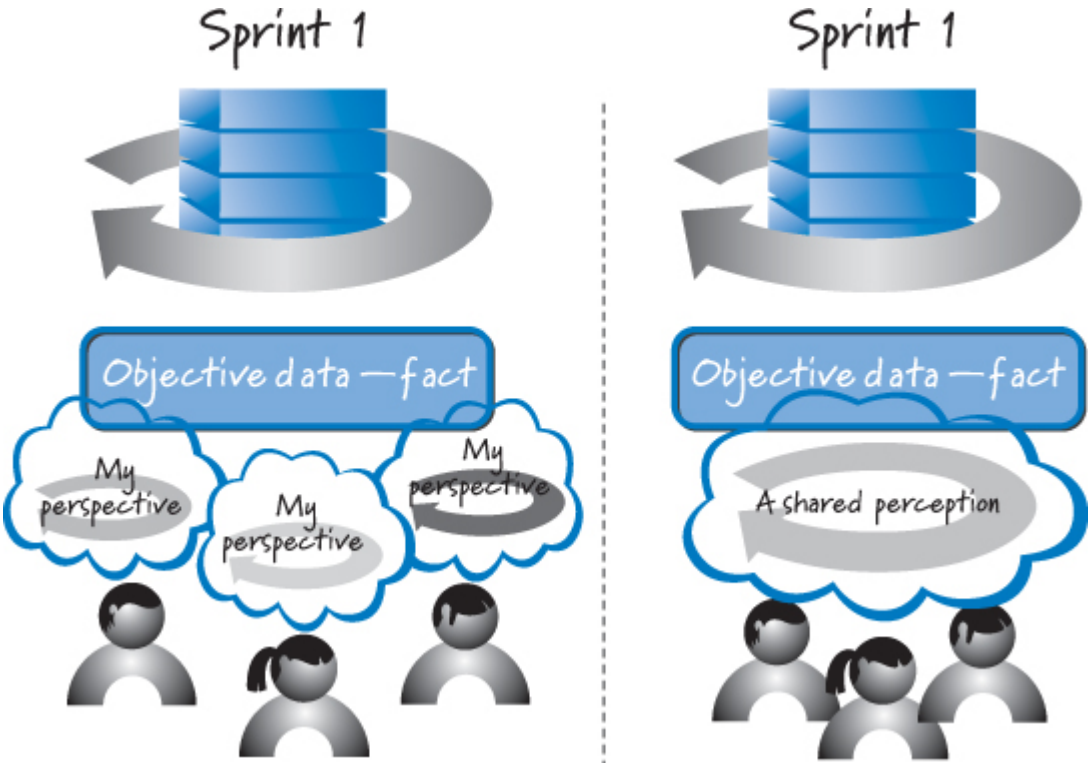

Figure 22.5. Aligning perspectives to create a shared context

The left side of Figure 22.5 shows that each person might view the sprint differently based on her own experience during the sprint rather than have a more big-picture view of the sprint events, accomplishments, and shortcomings. If individual perspectives are allowed to dominate, the retrospective could degrade into a session of opinion debate rather than a session focused on actionable outcomes based on a shared context.

When establishing a shared context, therefore, it's imperative that you first ground the retrospective in an objective, big-picture view of the sprint. After getting everyone in a talking mood, share objective data, such as committed PBIs, PBIs completed, number of defects, and so on. (Exactly what specific objective data is relevant should be based on the retrospective focus.) While most of the objective data is typically gathered during the prework, some objective data can also be left for the participants to collect collaboratively during the retrospective. Doing this can help energize the team around the importance of that data. Whether done as part of the prework or as a group, gathering objective data is crucial for establishing a common foundation built on facts rather than opinions.

Just because we are grounded in objective data doesn't mean subjective data is irrelevant, however. Each person brings to the retrospective subjective data reflecting her interpretation of the sprint. If that subjective data is not exposed and discussed, participants might just assume that everyone else experienced the sprint in a similar way. This misalignment will make it difficult for people to understand one another's comments and suggestions.

There are a number of exercises that the participants can use to develop a shared context of both objective and subjective data. Two of the most common exercises are an event timeline and an emotions seismograph.

Event Timeline

Creating an **event timeline** is a simple yet powerful way to generate a shared artifact that visually represents the flow of events during a sprint. Events could include "Busted the build," or "Interrupted to fix production failure," or "Salina returned from holiday."

A common approach is to draw a timeline on a wall or whiteboard and have the participants put cards (or sticky notes) on the timeline representing meaningful events that occurred during the sprint (see Figure 22.6). Distributed teams could do the same exercise using an online-shared whiteboard.

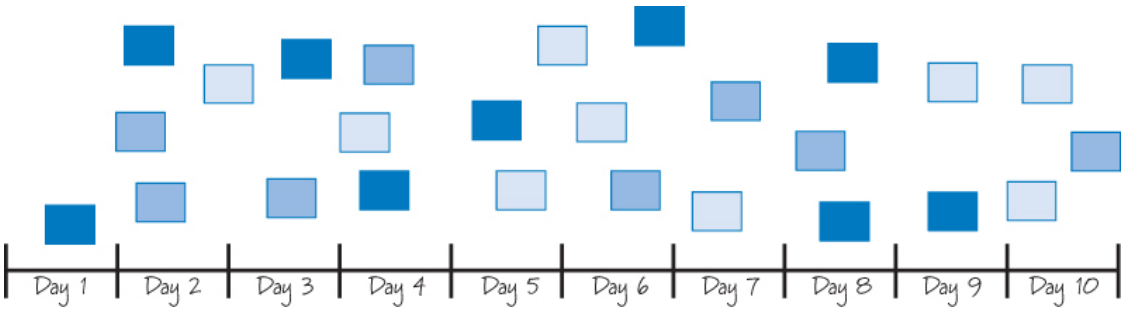

Figure 22.6. Sprint event timeline

The event cards are placed on the timeline in chronological order. This temporal view of events provides excellent visibility into the flow of ac-

tivities during the sprint and also provides a context for quickly identifying missing or forgotten events.

To help visually categorize events, many teams use a variety of colored cards. Some do this to represent different event types (for example, green is a technical event, yellow is an organizational event, red is a personal event). Other teams use colors to represent feelings or energy levels (for example, green is a positive event, yellow is a neutral event, and pink or red is a negative event).

Emotions Seismograph

Many teams create an **emotions seismograph** as a complement to their event timeline. This is a graphical representation of the emotional ups and downs of the participants over the course of the sprint (see Figure 22.7). Creating an emotions seismograph helps expand the shared context beyond the objective data (what happened) to include some subjective data (how the team felt about it).

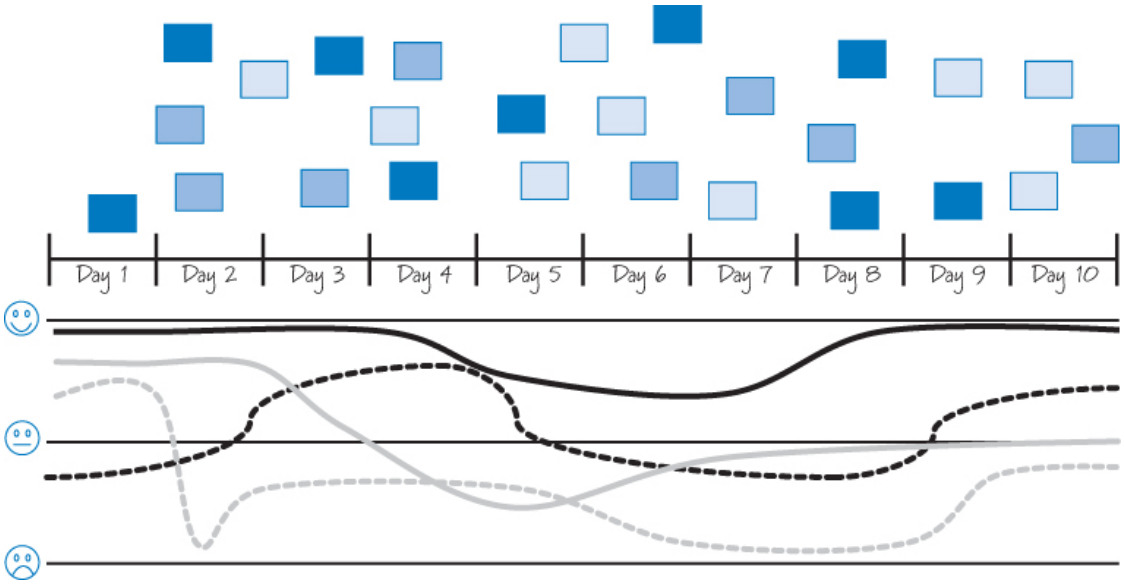

Figure 22.7. Emotions seismograph

To create the seismograph each participant is invited to draw a curve showing how she felt or what her energy level was like over the course of the sprint. It is frequently convenient to draw the seismograph directly under the event timeline so that the two sets of data can be visually corre-

lated. Later, the participants can mine this data for interesting insights for process improvement.

Identify Insights

Once a shared context has been established, the participants can thoughtfully examine, understand, and interpret the data to identify process improvement insights. Doing this effectively requires a system-level (bigger-picture) focus. Focusing on just one aspect (having a more localized view) might cause teams to miss the bigger picture. A system-level focus also helps teams move past the superficial and identify the root causes of issues.

The participants should start by mining the shared context data. For example, they could look at their event timeline and emotions seismograph and ask the following questions to help uncover insights:

- What worked well?
- What didn't work well?
- Where are some opportunities to do things differently?

Frequently participants are asked to brainstorm insights and then capture them on cards and place them on a shared wall or other surface so that everyone can see them (see Figure 22.8).

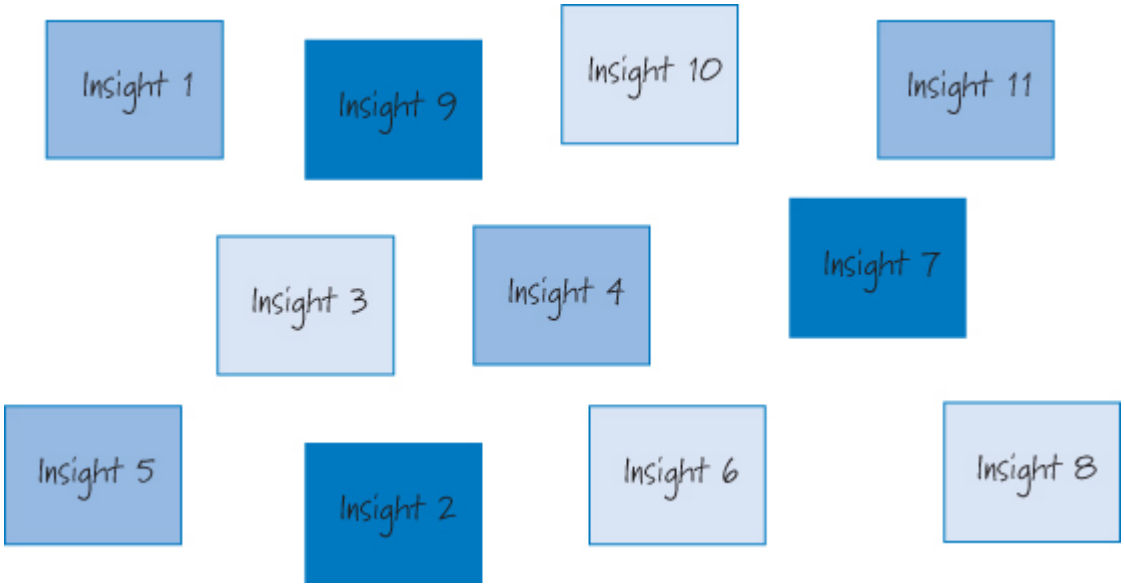

Figure 22.8. Retrospective insight card wall

Another source for insights might be the team's **insight backlog**, a prioritized list of previously generated insights that have not yet been acted upon. If such a backlog exists, mine it to see which insights the participants would like to include and consider during the current retrospective. Any existing insights should be represented by cards and placed on the wall alongside the new insights.

Once the cards have been placed on the wall, the participants will need to organize them. To do this, many teams choose an exercise such as **silent grouping** to cluster the insights into meaningful groupings to indicate similar or duplicate cards (see Figure 22.9).

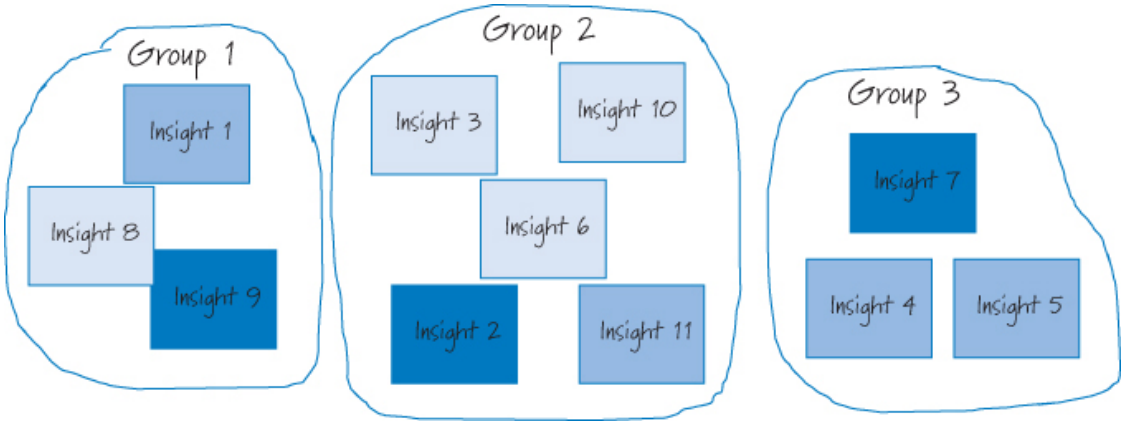

Figure 22.9. Insight cards clustered into similarity groups

During silent grouping, people collaboratively create the groupings without verbal discussion, relying only on the individual placement and movement of cards as a means of communicating and coordinating among the participants. Silent grouping is time efficient and effective.

Other teams prefer to divide the wall into category areas (such as things to keep doing, things to stop doing, things to try) before the retrospective begins. Then, as the insight cards are being created, the participants can place each card on the wall in the appropriate category (see Figure 22.10).

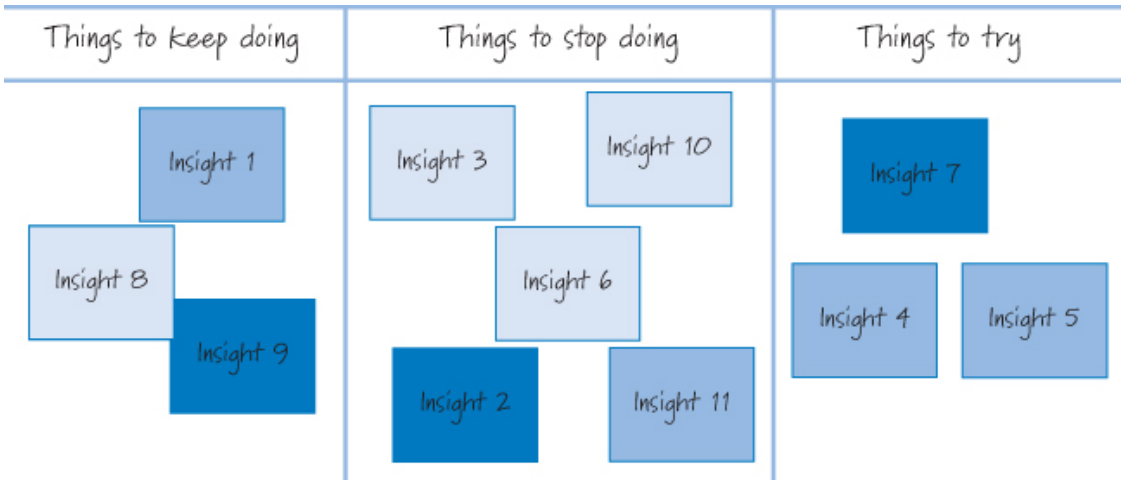

Figure 22.10. Insight cards placed into predetermined groups

Even with preassigned categories, however, it is still effective and efficient to have the participants perform silent grouping to cluster similar cards within a category.

After creating a shared context and mining the data for insights, the participants should have identified many areas for improvement in their use of Scrum and, by extension, the way they work together to deliver value. Some of these insights might lead to deeper discussions among the participants to better understand underlying causes, or important patterns or relationships. When all the insights have been discussed and organized on the wall, it's time to determine what to do with all of the information.

Determine Actions

Insights are our ideas or perceptions of things that can be improved. To extract long-term value from these insights we need to move from discussing them to taking demonstrable actions to leverage them. For example, if the insight is "We're wasting too much time because the code management system keeps failing," the improvement action moving forward might be "Have Talya apply the vendor patches to the code management system to make it more stable." Talya, a member of the development team, can take this action in the next sprint.

The participants should also take time to review what happened to the improvement actions from the last retrospective. If those actions have not been completed (or even started), the participants need to know why before they start addressing new insights. They might choose to carry forward previous actions or prioritize them against the new insights they have just identified.

Selecting Insights

It is important to realize that retrospectives frequently identify many more improvement insights than the Scrum team and organization can digest and act on in a short period of time. So, the participants first need to determine which improvement insights to act on immediately and which can be deferred. Many teams have the participants prioritize the insights based on what they believe is most important or where they are most energetic about seeing improvement. Sometimes these two are not the same. We might agree that a particular improvement insight is important, but if there is no appetite to do the work required to leverage the insight, it might not be a good choice right now. If the participants are energetic about an insight, they are much more likely to take concrete actions to leverage it.

One popular way to prioritize insights is to use **dot voting**, as illustrated in Figure 22.11.

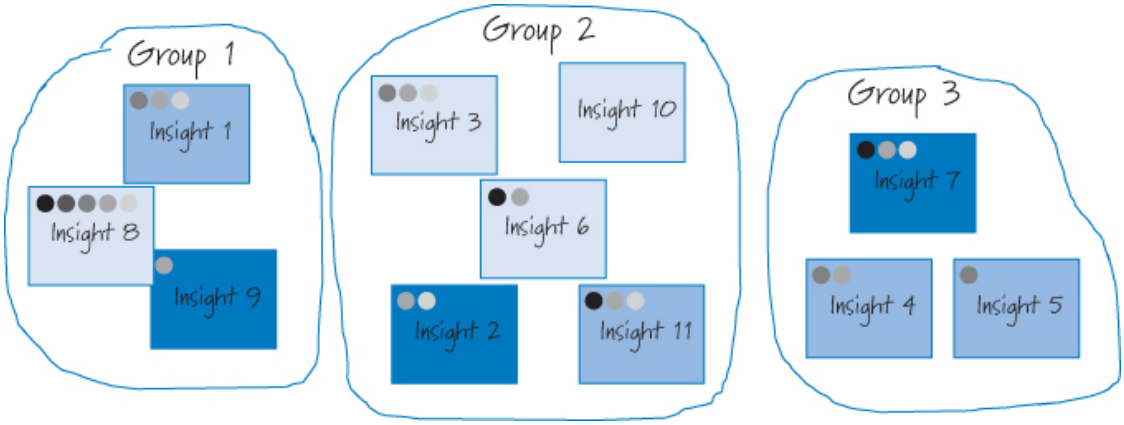

Figure 22.11. Example of dot voting

During dot voting, each participant is given a small number (perhaps three to five) colored dots. The participants then simultaneously place their dots on the improvement insight cards that they feel are the highest priorities to address. A person can put all of her dots on one card or spread them out over several cards. Once everyone has voted, the cards with the greatest number of votes should be considered first.

Exactly how many insights should the participants select to work on? Well, that depends on how much capacity the participants are able to dedicate to the insights and over what period of time.

Typically the period of time is the next sprint. So if the Scrum team is doing two-week sprints, it will likely consider insights that it can address over the next two-week period. Even if an insight is too large to be fully addressed in the next sprint, the participants might choose to start working on it and make demonstrable progress against it.

The participants must also determine how much capacity they can dedicate to addressing insights during the next sprint (or whatever time period they are considering). If the team plans to dedicate time in the next sprint to actions from the previous retrospective, it will clearly affect the team's capacity for new actions identified in this retrospective.

Spending time working on insights, old and new, leaves less time for working on features. So, how much time should the team allocate today to address insights that can provide a larger payoff later? Answering that

question really requires input from the product owner, which is one of the reasons it is important to have the product owner present at the retrospective. If the Scrum team doesn't specifically allocate time to work on improvement insights, a likely outcome is that insights won't get worked on.

Once we know the available capacity to work on insights, the participants can get a rough idea of which of the high-priority insights can be immediately addressed. However, the final decision can really be made only once the specific actions are determined.

Decide on Actions

At this point we have prioritized our insights and have some idea of the capacity we have to work on them. However, we get no measurable value from the retrospective until we define concrete, actionable steps to leverage our insights and improve the Scrum process.

Most actions will take the form of specific tasks that one or more Scrum team members will perform during the upcoming sprint. For example, if the insight is "It takes too long to determine when the code build breaks," the action might be "Have the build server send an email when the build is broken." This action requires task-level work on the part of one or more development team members. The team should determine who can do this work and how much time the work will take. Only then can the team be sure that working on a particular insight is doable within the available capacity.

Not all insights require specific task-level work. For example, an insight like "Be respectful to one another and show up to the daily scrum on time" should require little (if any) task-level work by the team. While there is a real action, "People should actually make the effort to show up on time," it will not reduce the team's capacity.

Sometimes the actions represent impediments that the ScrumMaster might own but someone else in the organization has to resolve. For exam-

ple, the insight might be "We can't get PBIs done because we need the latest version of a third-party vendor's software to test against." The action might be "Nina will work with our procurement department to obtain the latest vendor update." So Nina, the ScrumMaster, will work with the procurement people to address the third-party vendor issue that is impeding the team from getting PBIs done. This action will start in the next sprint, will need some capacity from the ScrumMaster, and may require several sprints to be resolved.

When determining the proper actions, we need to remember that it might not be possible to immediately address the insight. Instead, we might need to *explore* the insight before we can actually make an improvement. In such cases the proper action might be to investigate and collect data during the next sprint so that we can better understand the problem.

For example, the insight might be "We're puzzled by why two components that are fully tested and have their own automated test suites fail when they are combined into a cross-component automated test suite where each component is still individually executed." At this point there isn't a specific action the Scrum team members can take to address this insight because they really don't understand what is going wrong. However, the team can create an action for specific team members to explore this issue in the next sprint, and the team can determine how much capacity to allocate for exploration.

Insight Backlog

As I mentioned earlier, many teams create an *insight backlog* (sometimes called an *improvement backlog*) to hold any issues that are identified during a retrospective but cannot be worked on immediately. The idea is that at the next sprint retrospective the participants can choose to use the insights in the backlog as candidates to be prioritized against new insights when determining where to focus time in the next sprint. Of course, the insight backlog should be groomed periodically to ensure that its contents remain valuable insights.

Other teams simply discard any insights that they choose not to work on in the next sprint. The thinking is that if an insight is truly important, it will be identified again at the next retrospective.

Close the Retrospective

Once the final improvement actions have been determined, the participants close out the retrospective. Many close by recapping what actions the team has decided to take based on what the participants learned. This might be as simple as describing each committed action item and who is going to work on it.

Closing is also a good time to appreciate people and their participation. Each participant should say a few kind words of appreciation regarding the contributions made by others. Be sure to also recognize any non–Scrum team members who took time out of their busy schedules to participate in the retrospective.

Finally, it's a good idea to spend a few minutes asking the team for suggestions to improve the team's approach to performing a retrospective. A retrospective is, after all, part of the Scrum framework and as such should also be subject to inspection and adaptation.

Follow Through

To ensure that what happens in the sprint retrospective does *not* just stay in the sprint retrospective, the participants should follow up on the actions they chose to complete. Some actions (such as that everyone shows up on time for daily meetings) need only to be reiterated and reinforced by the team members and the ScrumMaster. Others will need to be addressed during the forthcoming sprint-planning activity.

Frequently the easiest way to handle the improvement actions is to populate the sprint backlog with tasks corresponding to each action prior to bringing in new features. The team's available capacity to work on new features would then be adjusted downward by the estimated time these

improvement tasks will take. Honestly, any approach that allows the team to make a good commitment at sprint planning while at the same time affording it the opportunity to work on the improvement actions is a good approach.

One approach that does *not* work is to have an "improvement plan" for the team that is separate from the work it will do each sprint. This two-pronged approach will almost always lead to the improvement plan being subordinate to the typical feature-driven sprint plan. To ensure that the improvement actions do take place, don't separate; integrate!

Actions that do not require team member time will likely find a home on the ScrumMaster's impediment list. And actions that are destined for other teams or the organization as a whole can be placed into the appropriate backlog for the people who are expected to do the work; the ScrumMaster typically follows up with the external parties to help ensure that these actions actually get done.

Sprint Retrospective Issues

Sprint retrospectives are not without issues. Having worked with many organizations using Scrum, I have noticed a number of common issues (see Figure 22.12).

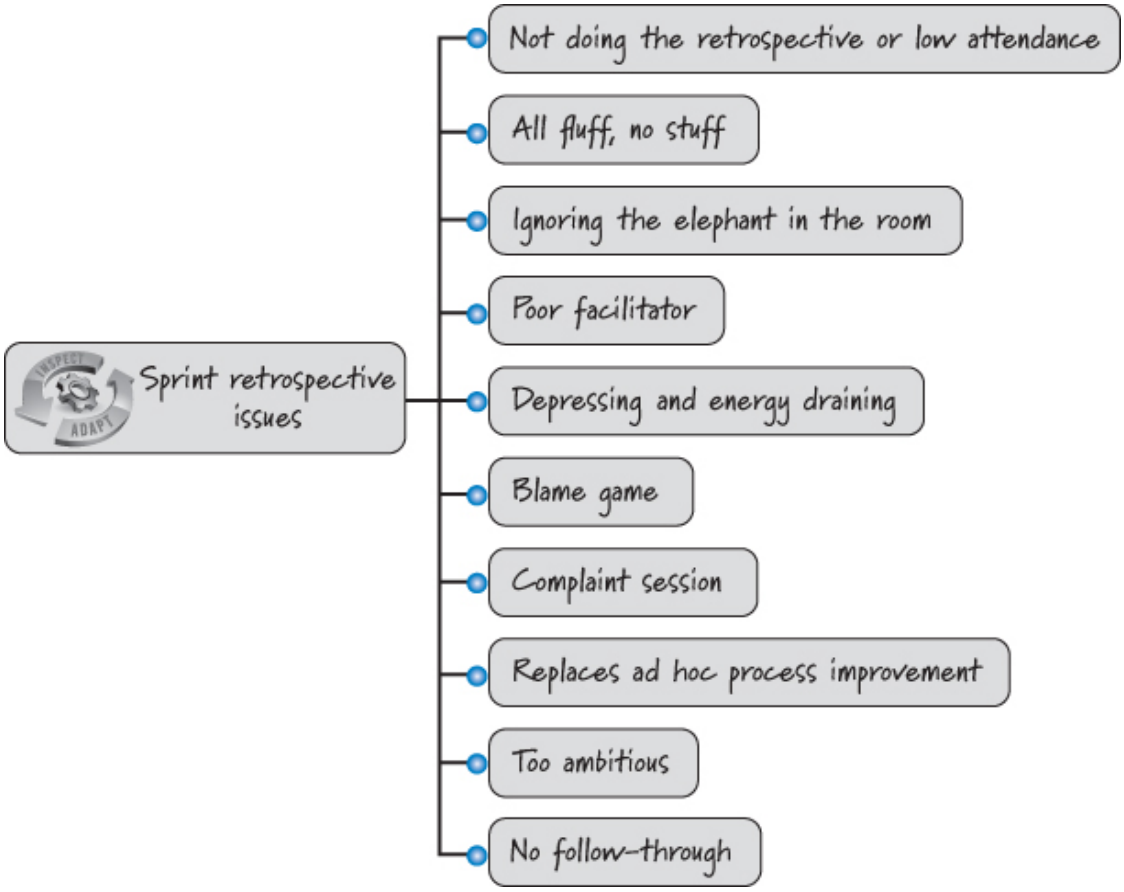

Figure 22.12. Sprint retrospective issues

An unfortunate issue is when teams simply don't do the sprint retrospective, or when they do, attendance is low. The reasons for both tend to be similar. If people are assigned to multiple teams, scheduling conflicts could prevent them from attending. This is an organizational dysfunction that managers need to address. Or perhaps team members are just bored or disengaged, or they have not really bought into using Scrum. Others might think that doing anything other than their particular work isn't worth their time (for example, they believe that anything other than coding or testing is just wasteful). Often these issues stem from naiveté regarding Scrum and its focus on continuous improvement. Other times it's just the opposite—team members believe they have reached the pinnacle of Scrum usage and therefore have nothing further to learn from the sprint they just performed, from their teammates, or from their own success or failure. If people don't see the value of doing a retrospective or of their attendance, consider dedicating some of or the entire next retrospective meeting to exploring this value issue.

Sometimes low attendance happens because it is inconvenient for remote participants to join by phone or video conferencing. If remote participants find attending the retrospective inconvenient because of when it is scheduled, consider changing or rotating the time so that no single location is always inconvenienced. If it is inconvenient because it is just hard to participate remotely, reconsider the current telecom infrastructure and how the exercises are being conducted to better incorporate remote participants.

Some retrospectives are very busy but really don't achieve anything actionable. I call these the *all fluff, no stuff* retrospectives. If all we get is fluff, we're wasting our time. Consider bringing in an outside, experienced retrospective facilitator to help the participants get to the real stuff.

Other retrospectives are fascinating to observe. There is clearly a critical issue that is having a dramatic effect on the team, but nobody will even bring it up. To reuse the old adage, the participants are ignoring the elephant in the room. There is probably a safety issue that is preventing people from discussing the elephant. The ScrumMaster should take a leadership role in helping the team and organization address the safety impediment first.

Other times the retrospective is just poorly facilitated. The facilitator, perhaps a new ScrumMaster, is trying her best but it's clearly not working. Perhaps an outside facilitator should be used for a few retrospectives.

Some retrospectives are downright depressing and energy draining. Perhaps the sprints are not going well and people view the retrospective as an activity that just compounds the misery by making them relive it. Consider spending a bit more time to set the appropriate atmosphere at the start of the retrospective. Also, an outside facilitator might be more effective at helping people stay focused on positive improvements.

Frequently retrospectives are depressing because people start playing the blame game and finger-pointing. The facilitator must extinguish this be-

havior as soon as it occurs to prevent a cascade of finger-pointing.

Other times a retrospective can degrade into a complaint session. Perhaps some people view it as therapeutic to just come and complain about the way things are (or at least how they perceive things to be). They have no desire to improve, just the desire to complain. Consider inviting people to the retrospective who can actually effect real change. Then have a face-to-face dialogue with them instead of complaining in their absence.

Another unfortunate situation is where participants consider the retrospective to be *the* time to do process improvement, thereby diminishing ad hoc process improvement during the sprints. A retrospective is a great time for the team to reflect on a period of work and discuss how to make things better, but it was never intended to be the replacement for ad hoc process improvement. The ScrumMaster should proactively promote healthy ad hoc process improvements throughout the sprint.

Sometimes our desires are bigger than our abilities. New teams that are energized and focused on really getting better can frequently become overly ambitious and set improvement goals that are totally unrealistic. Doing so will only lead to a big letdown when the team fails to meet its ambitious goals. The ScrumMaster should be vigilant and remind the participants of their available capacity to do improvements and help them moderate their ambitions.

Perhaps the biggest issue of them all is when there is no follow-through to actually work on the improvement actions identified during the retrospective. If we're not going to follow through, there's no need to waste our time on retrospectives. The ScrumMaster has a leadership role in helping the team constantly improve its process. If there is no follow-through, the ScrumMaster needs to be aggressive about working with the team to identify the root cause and helping team members address the impediment.

Closing

Sprint retrospectives are a time for the team to reflect on how well it is using Scrum and to propose improvements. The retrospective is a collaborative activity among the Scrum team members (and any non–Scrum team members on an as-needed basis). Once the retrospective prework is completed, the basic flow of the retrospective is to set the atmosphere to have a successful retrospective, get everyone on the same page by creating a shared context grounded in data, identify improvement insights, determine improvement actions, and then close the retrospective. After closing the retrospective, it is critical that the participants follow up and carry out the improvement actions so that the team is more effective during the next sprint. It is also important to keep a watchful eye out for issues that might prevent the retrospective from being successful and to quickly act on them.