Chapter 8

RETROS, REVIEWS, AND RISKS

Thankfully, the days of the onerous, postmortem debrief (held at the very end of a long project when it is least helpful) are over! Instead, our Scrum projects opt for frequent inspection and adaptation of both the product and the process at the end of every sprint to ensure that risks are mitigated and lessons are being applied when they matter most.

The three shortcuts in this chapter provide suggestions for making the most of your regular sprint reviews and retrospectives.

Shortcut 22: To-Dos for Your Sprint Reviews focuses on steps to take to ensure your team doesn't sell itself short during the open sprint review. Shortcut 23: Retrospective Irrespective suggests two effective retrospective techniques as well as key areas to focus on during this end-of-sprint activity. Shortcut 24: Risk Takers and Mistake Makers discusses the need to generate a failsafe environment, which in turn generates a culture of openness and innovation.

Shortcut 22: To-Dos for Your Sprint Reviews

WARNING: On the surface, this Scrum activity appears to be both simple and straightforward. But be warned: without careful preparation, this session can lead to riotous table-thumping and streams of tears.

Just show the stakeholders what was completed over the last 2-week sprint—sounds simple right? Well, based on my experience, a sprint review is rarely simple, and in fact, I consider it to be the most delicate session to facilitate.

The core issue is aligning the expectations of a disparate group of stakeholders outside of the team. These people are often more senior in the business relative to the team, they most certainly have less familiarity with the project compared to the team, and without wishing to generalize, they often have the attention span of a puppy with an attention-deficit disorder!

During Sprint Planning

Preparation for the sprint review actually starts during the previous sprint planning (see Figure 8.1). During the planning session, the team must ensure that tasks are being allocated appropriately in readiness for the sprint review. These tasks include

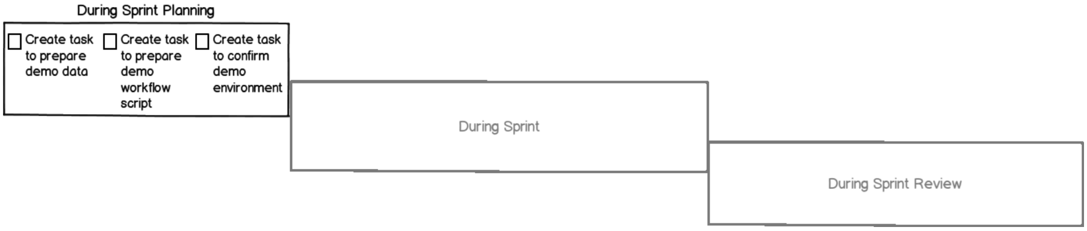
During Sprint Planning
□ Create task to prepare demo data
□ Create task to prepare demo workflow script
□ Create task to confirm demo environment
During Sprint
During Sprint Review

FIGURE 8.1 Preparation for the sprint review actually starts in the previous sprint planning session.

- Preparing some basic demo data
- Preparing a demo workflow script
- Ensuring the demo environment is working as expected

You don't want the team to spend too much time on these tasks, as the sprint review shouldn't be turned into a dog-and-pony show, but these tasks certainly need to be acknowledged.

There are also a few important points to stress to the team during sprint planning. First, the sprint review demo needs to be conducted on the staging environment rather than on the development server (and definitely not on an individual's machine). This is not a smoke-and-mirrors demonstration, and the team needs to really prove that what it has developed is releasable.

Second, although the team should feel relaxed during this session, it still needs to be taken seriously. During the sprint review, the efficacy of both the team and Scrum is on display to those who might not have had much exposure to either.

During the Sprint

Once into the sprint, the ScrumMaster and product owner need to start running through their collective checklist in anticipation of the review (see Figure 8.2). Items to cover off include location, invitations, painful attendees, presenters, and expectations.

Location

Confirm the location of the sprint review. Make sure that any relevant meeting rooms have been booked and that they are well equipped for the occasion. Double-check that a suitable projector, ample seating, network connectivity and a whiteboard are available. I also recommend testing that the equipment works as expected; nothing is more embarrassing than so-called technology experts unable to get the projector to work before the session even begins.

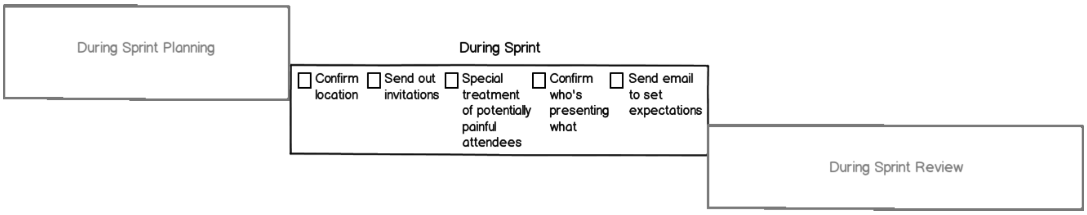
During Sprint Planning
During Sprint
□ Confirm location
□ Send out invitations
□ Special treatment of potentially painful attendees
□ Confirm who's presenting what
□ Send email to set expectations
During Sprint Review

FIGURE 8.2 During the sprint, the ScrumMaster and product owner should go through their collective checklist in anticipation of the review.

Invitations

Ensure that invitations are sent to the relevant stakeholders as early as possible. In fact, there should be a standing invite to the regular attendees, so this suggestion applies only to any anomalous extras.

Painful Attendees

You may have concerns about a particular stakeholder who has a habit of taking center stage and issuing unwarranted criticism or off-tangent comments. If you have one of these delightful characters to deal with, I recommend that the ScrumMaster and/or product owner visit him or her before the sprint review to present a sneak peak and to get some early input. People who feel a little bit special are less likely to be destructive during the review. We don't want to fix the results, but we certainly want to avoid unnecessary morale-busting reviews.

Presenters

Near the end of the sprint when you are able to ascertain exactly which user stories will likely be demonstrated, ensure that the team has decided who will be demonstrating what. It might be just one individual, or it could be a group effort—there is no hard and fast rule.

Expectations

Once the demonstrable stories have been determined, I recommend circulating an email to all attendees to give them a brief heads-up of what is going to be presented. The reason for doing so is that in the early sprint reviews of a project, the user-ready features with a snazzy UI are typically few and far between, which can lead to some miffed, eyebrow-raising stakeholders. For example, by informing them that the upcoming sprint review focuses primarily on the less visual, technical "plumbing," there is less likelihood of misaligning expectations.

This email also serves as a friendly reminder of when and where the session is with a gentle warning that any latecomers will have to supply coffee!

During the Sprint Review

When it comes time to conduct the actual sprint review (see Figure 8.3), consider the following tips to help ensure a smooth ride.

Refreshments

Everyone loves snacks. Ensure that there is something to nibble on as well as drinks (hard whiskey if you're really worried about how the review will go). I find it amusing that people's positive or negative judgment of a conference or training session often comes down to the catering, so don't forget to look after everyone's stomachs!

Scene Setting

Once everyone has settled in and engaged in some light banter over the refreshments, the product owner should briefly outline the sprint goal as well as reiterate what will be demonstrated. Also, for early sprints, it doesn't hurt to explain the definition of done (see Shortcut 11) to the stakeholders.

Impediments

Following the scene setting, it can also be a good idea to discuss any impediments that impacted the sprint, including why they occurred and how they were dealt with. This is an opportunity to lobby for greater assistance if there are any systemic impediments. An example might be the physical environment—perhaps a problem dealing with the facilities department in your mission to get larger desks or more breakout space.

In addition, I recommend using this segment to briefly make a point of key process improvements that were implemented to help achieve the sprint goal. Don't go into detail here, as this session is about reviewing the product, but a quick mention won't hurt and is a great opportunity to demonstrate to the stakeholders that the team is constantly improving.

The Main Event

Instead of a one-way showcase, the demo of the sprint's completed work should act as a prompt to encourage a two-way conversation between the business and the Scrum

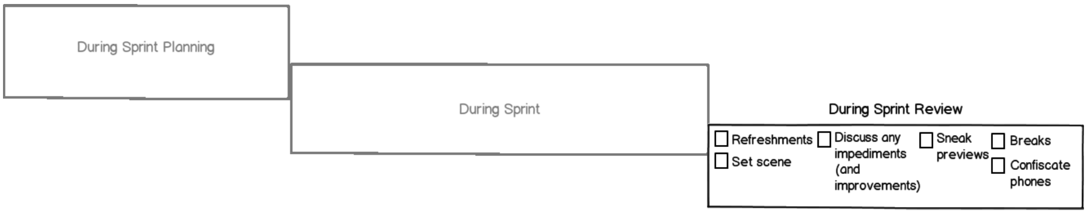

FIGURE 8.3 Some important points to remember during the sprint review.

team. It should be an open and honest discussion focusing on what was completed and what is coming up next.

Remember that this session should not become a smoke-and-mirrors slide show presentation to impress the attending stakeholders (unless, of course, your team is actually developing PowerPoint[1] or Keynote[2]!). I assure you that misleading demonstrations will only come back to bite the team.

Preview at the Review

It is a fundamental tenet that during the sprint review, the team should demonstrate only stories that meet the definition of done. It makes sense, but it can be very frustrating for some stakeholders, and the reality is that the team will possibly receive pressure to still show what has been worked on (even if it is not quite finished).

In this situation, instead of being a stubborn mule, I recommend creating an additional agenda item labeled "Coming Soon." This way, much like a movie preview, there is acknowledgment that the work isn't complete, yet the stakeholders still get a sense of work that's on the boiler and is coming soon to a sprint review near you!

Breaks and Phones

If your session is longer than an hour, I suggest taking 5-minute breaks every 45 minutes to maintain focus. But, be aware of the attention-deficit-disordered puppies who will get easily sidetracked and mugged in the corridors: don't let them stray far.

Also, although tough to enforce, try requesting that everyone submit their BlackBerrys[3] (providing that RIM is still around when this is published) at the door. The only problem is that you may need to hire armed guards to pry those smartphones out of stakeholders' hands! At the very least, you should announce at the start of the meeting, "For the safety of this sprint review, we request that you turn off all electrical devices." Good luck and at the very least, think up a creative punishment for those whose phones ring during the session.

So-Called Suggestions

There will no doubt be a range of questions and suggestions that surface throughout the session. I recommend that questions be controlled and kept on topic. The team should answer any and all questions that surface in relation to what is being demonstrated; however, questions that are tangential or completely off topic should be taken "offline" by the product owner and discussed with the relevant stakeholder(s) in a separate meeting.

[1] To find out more about Microsoft PowerPoint, go to http://en.wikipedia.org/wiki/Microsoft_PowerPoint.

[2] To find out more about Apple Keynote, go to http://en.wikipedia.org/wiki/Keynote_(presentation_software).

[3] To find out more about BlackBerry, go to http://en.wikipedia.org/wiki/BlackBerry.

Acknowledge any suggestions made (no matter how outlandish they might sound) by writing them down on the whiteboard or index cards. With any luck, after seeing any crazy suggestions written in black and white, stakeholders will retract their pearls of wisdom. Any reasonable suggestions should of course be added to the product backlog (by the product owner) for further assessment and consideration.

Picnics or Battles

Sprint reviews can be akin to turning up for a summer picnic or, alternatively, turning up for pitched battle! It comes down to taking the necessary precautions and treating each session seriously while still having some fun. Don't assume that every stakeholder has the same level of background as the team, and ensure that all attendees are made to feel comfortable by always explaining what is happening and why.

Aligning everyone's expectations is the name of the game, and achieving this objective is critical if you prefer picnics to battles!

Shortcut 23: Retrospective Irrespective

Unfortunately, when a team is under pressure, the important sprint retrospective is often the first to get bumped until life is a bit more relaxed and under control. Even I have to admit that, in my earlier days as a ScrumMaster, I was guilty of skipping the retrospective on occasion. The irony is that this session is most valuable when the pressure is on and/or when things aren't running as expected. As such, discipline needs to be maintained, and the sprint retrospective should *always* remain a consistent event at the end of each sprint, irrespective of the noise surrounding it.

Reinforce Scrum's Values

It is important to remember Scrum's core values: *openness, courage, respect, focus*, and *commitment*. Nowhere should these values be applied more rigorously than in the sprint retrospective. Without an atmosphere of openness, you will never get to the heart of issues; without courage, the team won't be willing to confront problems head on; without respect, the team won't present criticism in a constructive fashion; and without focus and commitment, the team won't care enough to resolve the issues. Everyone should keep these values in mind at all times.

What If We're Running One-Week Sprints?

I've been asked before by teams who are running in very short, 1-week sprints whether they should perhaps run a retrospective only every two or three sprints (as opposed to on a weekly basis). Personally, I feel that 1-week sprints are too short (see Shortcut 8); however, if there is an insistence on shorter sprints, I don't recommend skipping the retrospective sessions. Instead, simply run shorter retrospectives but make sure that they still occur every sprint.

Location, Location, Location

To create an atmosphere of openness, the location where you hold your sprint retrospective is important. A large, stuffy boardroom with a long, impersonal mahogany table isn't conducive to the atmosphere you need and may turn your retrospective into more of an inquisition. Instead, I recommend a relaxed setting such as a coffee shop, a break room (if you have one), or even the kitchen area, which, because of its proximity to snacks and coffee, can be a great choice!

Getting Set

Ideally, you want your team arriving for the retrospective session prepared with a list of issues and suggestions so that you can get straight into it. To help facilitate this preparation, I recommend that you send an email before the session outlining potential areas to focus on. Examples of key areas include

- Communication
- Processes
- Scope
- Quality
- Environment
- Skills

Let's now explore a sample of some common improvement actions that might be triggered in each of these areas.

Communication

Communication areas include the following:

- Fixing disjointed communication between the product owner and development team (especially if the product owner isn't seated among the group).
- Removing disruptive and unnecessary communication between external stakeholders and the development team.
- Resolving dysfunctional communication within the team, typically due to an overreliance on written documentation (and email) rather than face-to-face discussion.

Processes

Process areas include the following:

- Upgrading software, hardware, and network connectivity.
- Confirming that the Scrum processes are well understood and clearly defined.
- Maintaining engineering standards relating to code quality, source control, and deployment processes.

Scope

Scope areas include the following:

- Making certain that significant changes to scope do not occur mid-sprint.
- Maintaining formatting consistency across all user story descriptions and acceptance criteria.
- Ensuring that the sprint planning scope is not ill-defined or too vague.

Quality

Quality areas include the following:

- Clearly defining and maintaining a consistent definition of done.
- Evolving test practices to enable more mature test automation.
- Ensuring that programmers are assuming ownership for the testing of their code and not leaving it totally up to the testers and/or product owner.

Environment

Environment areas include the following:

- Maintaining a collaborative environment that isn't too noisy and disruptive.
- Providing ample air-conditioning or heating.
- Making sure appropriate amenities are available (one microwave isn't going to cut it for 50 people!).
- Ensuring enough whiteboard space and other collaborative tools.

Skills

Skills areas include the following:

- Providing adequate training when new technologies are required.
- Conducting induction training for any new team members.
- Obtaining relevant specialist consulting when required.

Output of the Retrospective

When the juices start flowing, it isn't hard to generate a long list of issues to tackle. The trick is to not fall into the trap of getting overly keen and declaring that all issues will be resolved by the next retrospective! Instead, ensure that all improvement suggestions are documented, but aim to tackle no more than a few issues—perhaps one biggie and a couple of smaller ones. Nothing loses credibility faster than overpromising and underdelivering; so instead, do the opposite! Finally, write the agreed-upon actions on a large sheet of paper and place it near the task board as a constant reminder for the team. Please use discretion with this public display of action, as writing "Lock disruptive project sponsor in cage" and sticking it on the wall might not do much for your future job progression!

Format of the Retrospective

To keep things fresh, vary the format of the sprint retrospectives throughout the project. A number of approaches can be utilized, but for the sake of brevity, I focus on two of my favorites, nicknamed Circles and Bubbles.

Circles

The Circles technique employs an affinity mapping variation. For those unfamiliar with affinity mapping, it is essentially a graphical tool designed to amalgamate loose, unstructured ideas into collective groups.

Before reviewing the technique, let's identify the classic four questions often asked in Scrum retrospectives:

- What did we do well?
- What can we do better?
- What should we try next time?
- What issues do we need to escalate?

I prefer to simplify things even further by asking only two questions:

- What can we improve?
- What are we doing well?

Getting back to the technique, you first need to draw a horizontal scale on the whiteboard. On the very left write the label *Needs Improvement*; in the middle, jot down the more neutral *OK* label; and on the right, add the more encouraging *Going Well* label.

Now in keeping with the Scrum sticky-note tradition, hand each team member a bunch of sticky-notes and ask them to start writing their thoughts and/or issues on them. Once complete, direct the team to start sticking the notes on the wall, wherever along the scale they feel each note fits best. Be sure to time-box this step to avoid the dreaded analysis paralysis.

Once everyone places their notes on the board, it is time to align similar items to form groups. Next, use a nice thick marker (nonpermanent, or your issues will remain forever) to draw *circles* around the notes that reflect similar thoughts that are grouped closely together (see Figure 8.4); ignore the outliers for now. It will be rare for you not to identify a few common themes.

Once the groupings are completed, it is wise to briefly discuss any outliers with the team. More often than not, outliers result from misinterpretation or lack of understanding rather than from conflicting opinions.

Finally, as a team, address the circled groups in priority order (items further to the left will typically be of higher priority), and identify some actions to be followed up on in the forthcoming sprint to help address the improvement areas.

I like this approach for a few reasons:

- It avoids putting anyone on the spot (or in the spotlight).
- There is plenty of physical motion, which keeps the team alert and active.
- It is tactile and visual, which always makes life a bit more interesting.
- The team is able to immediately gauge the collective priorities by viewing the sticky-note groupings on the wall.

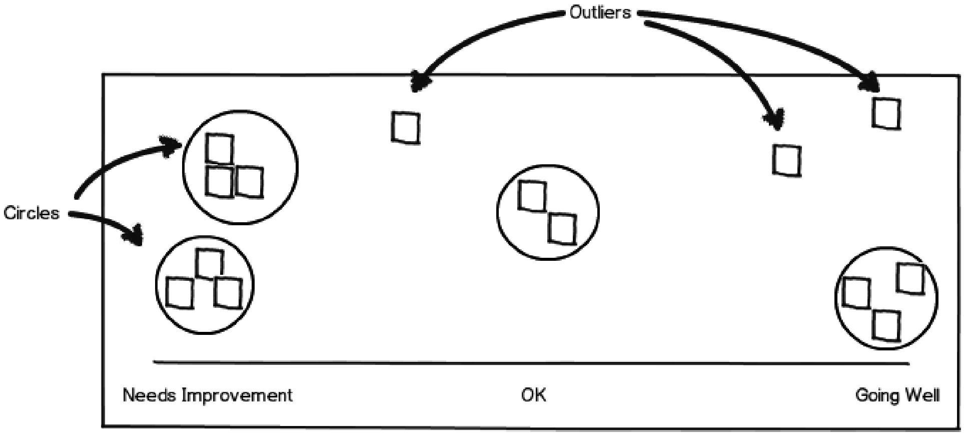

FIGURE 8.4 A Circles retrospective with similar suggestions grouped and circled.

Bubbles

The Bubbles technique is definitely my preferred approach for the first few retrospectives that you run with a freshly formed Scrum team (when the members haven't worked together before).

Bubbles works really well because it promotes immediate collaboration, helps everyone get to know one another in a noncontrived fashion, and again, doesn't put anyone on the spot.

So, how does it work? First, you need to come to the session prepared with paper and pen for everyone in the group.

This technique works best with an even number of team members, but you can easily make it work with uneven numbers. For demonstration purposes, let's say you have a team of eight.

The first step requires each person to document on paper the three most urgent issues that he or she feels should be focused on in the upcoming sprint. I recommend a 5-minute time-box for this step.

Next, get the individuals to pair up with a buddy (here you will obviously have four pairs, but there is no problem having groups of three instead if you have an uneven number). Again, during a 5-minute time-box, ask the pairs to decide, based on their combined lists, which three items should take priority overall.

You can probably see where this is heading by now, but in case you haven't guessed, the next step is to join two pairs together so that you now have two groups of four (see Figure 8.5). Once again, time-box this step, although you may offer a little

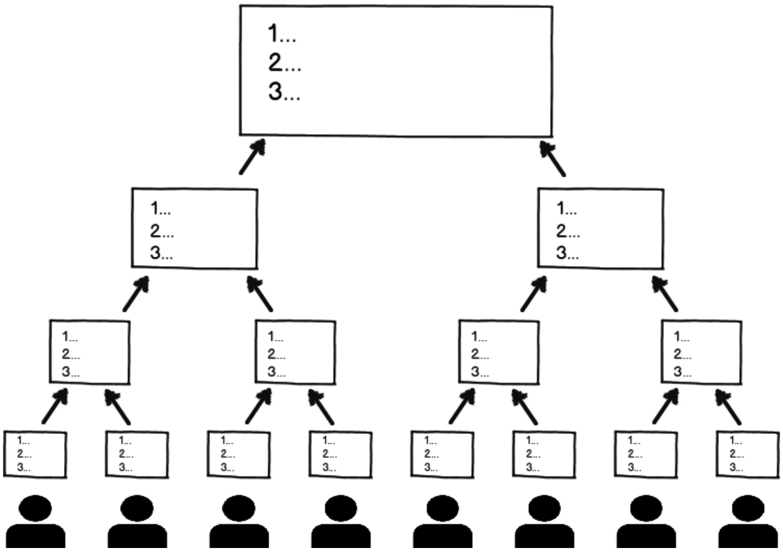

FIGURE 8.5 With this approach, the most urgent and important issues "bubble up" to the surface.

extra time because you now have groups of four debating the merits of their lists to determine the top three.

The penultimate step is to form a single group of eight to determine which three issues “bubble” right to the top of the top. By this stage, you should find that everyone’s inhibitions have been dropped and that a healthy, open debate can now ensue, following which you will have determined what should be focused on next sprint.

Don’t disregard all of the other valid suggestions that were uncovered in the earlier rounds of Bubbles. Instead, document them so that when you return for the subsequent retrospective, there is a healthy starting point to help get the ball rolling. Suggestions that were previously lower on the list may at a later stage become the top-priority focus points.

Finally, the team can rinse and repeat the same Bubbles activity to identify and acknowledge the successful behaviors and processes that the team wishes to explicitly keep active within the next sprint.

Seasoned Pros

When you are working with a truly battle-hardened group of Scrum aficionados, my advice is to forget about any formalized structure for your retrospective—it will just appear contrived. When a team is comfortable in its own skin, you will have no trouble getting the collective group to speak up and express their concern or joy with the status quo. Instead, simply go out for lunch or coffee with a pad and pen and shoot the breeze.

Retrospective Attendees

The broader Scrum community seems to be a little divided regarding the question of whether product owners should attend the sprint retrospective. Personally, I believe that they absolutely should attend because they form an integral part of the Scrum team. That being said, before your team becomes a well-oiled Scrum machine, there may be a few communication breakdowns, especially between the developers and the product owner. If you observe any tension, I recommend that in addition to the “official” retrospective session, the ScrumMaster should also conduct an informal one separately with the product owner.

Keep It Fresh

Inspection and adaptation is the name of the game, and the sprint retrospective is vital to ensure that your team is constantly improving. Numerous techniques can be employed for this session, so feel free to vary the techniques to keep life interesting and fresh. Esther Derby and Diana Larsen’s *Agile Retrospectives* (2006) offers a whole host of different options that you can employ.

Most important, whatever happens, don’t skip your retrospectives, irrespective of what is going on in the background!

Shortcut 24: Risk Takers and Mistake Makers

My mother is a retired kindergarten director without one iota of experience in the software world. That being said, she is a smart cookie with exceptional interpersonal skills, making her a fantastic educator. Now, while it would be a futile exercise discussing the technical elements of my profession (such as refactoring and code reviews) with my dear mom, I can certainly talk to her at length about the more communication-centric Scrum elements, such as sprint retrospectives and self-organizing teams.

I remember her reply when I was explaining some of the "revolutionary" ways that Scrum was assisting teams, such as the inclusion of frequent and open retrospectives. Her surprising response was, "What's the big deal? That just seems extremely obvious to me." At first I laughed at her apparently naïve comment, but after thinking about it further I realized that she was right! It absolutely should be obvious, so why isn't it? How come it is such a relatively new construct in our industry? After much thought, I concluded that the answer to that question comes down to one big ugly monster:

Fear!

It is no coincidence that one of Scrum's core values is courage. To successfully implement Scrum, we must overcome a range of fears. It is important to be fearful of fear (excuse the recursion), because it inevitably leads to diminished innovation, fosters a blame-game culture, and causes analysis paralysis when every action the team takes is scrutinized to avoid mistakes and the whipping stick that follows. With that scary prelude out of the way, let's tackle a selection of these common fears.

Fear of Change

This is a classic human phobia, and when we leap into Scrum from a more traditional approach, we make change in spades, don't we? How about programmers doing testing, releasable software every sprint, cross-functional and self-organizing teams, not to mention the lack of project managers, just to name a few—yikes!

So here's the thing: change is scary. It takes people out of their comfort zones, disrupts the status quo, and is often perceived as a threat to many who are comfortable with the present state. The sad reality is that there will always be those in any organization who will simply reject change irrespective of the expected benefits. It helps to understand and accept this organizational axiom, or you'll find yourself dwelling in a constant state of disappointment. I agree with Mike Cohn when he writes, "Rather than focusing on those who are reluctant or opposed to Scrum, spend your time and effort helping those who are already enthusiastic" (2009). I find that this is the best way to build the momentum needed to convince the naysayers.

Free to Change

A real key to initiating change is how the transition is framed and communicated. Not many folks are comfortable with forced or categorical change that completely readjusts the existing landscape. Instead, I recommend presenting the transition as an ongoing experiment. Communicate the fact that the change is under scrutiny rather than being a fait accompli, and if something doesn't work, the team will simply inspect and adapt together.

Understanding this failsafe option often allays the fears of those who are afraid to change.

Fear of Exposure

Some developers just like to be left alone in a corner to focus on their work (for better or worse) and can become quite defensive about perceived regular inspections of what they are working on. The problem for these closed-book developers is that Scrum is all about regular inspections—not to spy, of course, but to identify waste and misguided activity.

Inspection sessions such as the daily scrums, sprint reviews, and retrospectives may also uncover those individuals (and I use that word on purpose) who genuinely don't wish to do the right thing for the team. These few bad eggs are often the ones who are most afraid of exposure, fearing that the rug will be dramatically pulled out from under them and their poor-quality work will be discovered.

Free to Be Exposed

Growing up in Australia, we are taught very early on how dangerous sun exposure can be. Stay out too long, and you will be nursing red, painful skin for several days. That being said, let's not be too tough on our sunny source of life, as staying out for a short, disciplined period of time can lead to a nice, healthy glow and an important dose of vitamin D.

Scrum's focus on regular inspection of both the product and the process is all about gaining the glow and avoiding the burn. It is much easier to take care of tanned skin as opposed to red raw, burnt skin. This is exactly the same as in software development. If we detect issues early and often, there will be fewer painful surprises later on that require significant nursing and attention.

When inspections are framed in a positive way, team members (who genuinely wish to do the right thing) will embrace these sessions even if they have a natural inclination to stay in a corner with headphones on. And what about the bad eggs? Well, you will be able to find them nice and early so that they don't ruin everyone's breakfast!

Fear of Making Mistakes

Twisted organizations that have been warped and corrupted by a culture of finger-pointing, cover-ups, and internal conflict will have a very difficult time embracing Scrum or any other framework grounded in empirical process control. Scrum turns software development into an open book where mistakes are clearly visible.

The profound problem for these types of organizations is that the software industry (in particular) is a breeding ground for making mistakes because of the inherent risk surrounding developing software. Roman Pichler (2010) explains:

> Risk is therefore an intrinsic part of software development; no product can come to life risk-free. Correlated with risk is uncertainty. The more uncertainty there is, the riskier the project is. Uncertainty, in turn, is caused by a lack of knowledge. The less we know about what to develop and how to do it, the more uncertainty is present. Knowledge, uncertainty, and risk are therefore interlinked.

Free to Make Mistakes

The simple fact of the matter is that the greatest learning comes from taking risks and making mistakes. Don't believe me? Well, how about this example. Dan Senor and Saul Singer, authors of *Start-up Nation*, teach us about an aspect of Israel that is far more positive than the media images of constant conflict in its rough neighborhoods. The impressive revelation is that despite the ongoing constraints brought about by the constant threat of war and the fact that its population is a mere 7 million, it happens to have the highest number of technology start-ups in the world listed on the NASDAQ stock exchange (excluding only the United States and China). Senor and Singer (2009) offer some thoughts on why this might be the case:

> Israeli attitude and informality flow also from a cultural tolerance for what some Israelis call "constructive failures" or "intelligent failures." Most local investors believe that without tolerating a large number of these failures, it is impossible to achieve true innovation. In the Israeli military, there is a tendency to treat all performance—both successful and unsuccessful—in training and simulations, and sometimes even in battle, as value-neutral. So long as the risk was taken intelligently and not recklessly, there is something to be learned.

For me, this is a true lesson in macro-level agility, and it is reinforced by my own interactions with Americans and Israelis in the technology scene who certainly seem to have an amazingly high tolerance for so-called failure. As Asher Moses (2012) writes in the *Sydney Morning Herald*, "Over in Silicon Valley, failure is celebrated and seen as a chance to learn." We Aussies (among many others) can learn a lot from this positive attitude toward risk taking and mistake making.

I like the example set by software companies such as Facebook who often require developers to "push code to the live site within their first week" (Bosworth 2009).

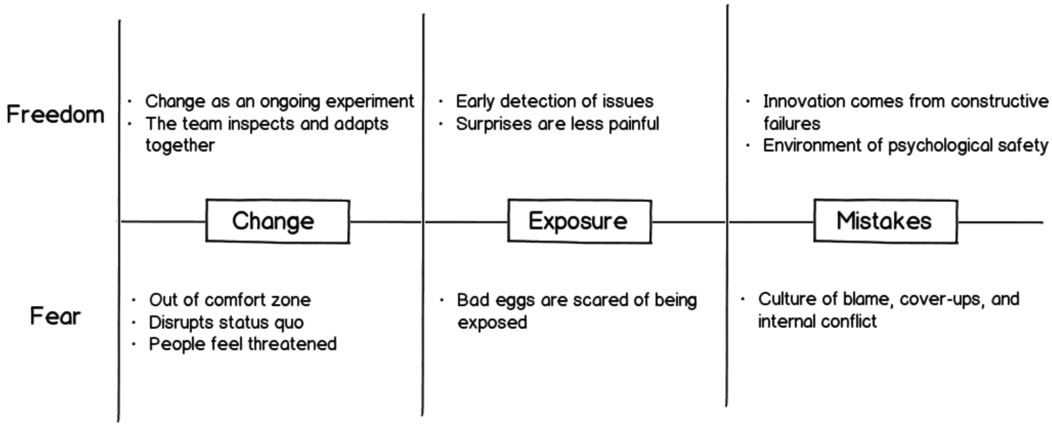
Freedom
· Change as an ongoing experiment
· The team inspects and adapts together
· Early detection of issues
· Surprises are less painful
· Innovation comes from constructive failures
· Environment of psychological safety
Change
Exposure
Mistakes
Fear
· Out of comfort zone
· Disrupts status quo
· People feel threatened
· Bad eggs are scared of being exposed
· Culture of blame, cover-ups, and internal conflict

FIGURE 8.6 Whether it's change, exposure, or mistakes, for every fear there is also potential freedom from the fear.

This is smart psychology. It indicates to newbie team members that they are not going to achieve perfection immediately, that there is almost the expectation that a mistake could happen, and that it isn't a big deal. Getting the first mistake out of the way early and in a safe setting creates an environment of "psychological safety" (Cohn 2009). It generates comfort and a breeding ground for calculated risk and innovation.

Figure 8.6 offers a quick summary of these fears and how teams can become free of them.

Lighten the Mood

Nothing beats fear more than lightening the mood. In a particularly volatile pressure-cooker environment that I once worked in, we had a CEO who would tangibly relax a team following a costly mistake by announcing with a chuckle that, "It appears that the f**k-up-fairy has been to town again, doesn't it, people?" Although a little crude for some, this simple expression seemed to always relax the room before a solution was rapidly figured out.

Another beneficial side effect that you're likely to experience when openly discussing mistakes and vulnerabilities is the forging of closer bonds between team members. This openness helps to break down the artificial armor of perfection that many of us wear to hide the fact that we are human and fallible.

Finally, remember that every cloud has a silver lining. Those who are able to turn mistakes and problems into lessons and opportunities will soon find a working life totally free from fear.

Wrap Up

The three shortcuts in this chapter focused on a selection of tactics, tools, and tips to help your team make the most of their sprint reviews and retrospectives. Let's recap what was covered:

Shortcut 22: To-Dos for Your Sprint Reviews

- How to effectively prepare for the upcoming sprint review
- The importance of not turning the sprint review into a one-way demonstration
- Elements to consider incorporating in the sprint review

Shortcut 23: Retrospective Irrespective

- The importance of never skipping a sprint retrospective
- Two useful retrospective formats: Bubbles and Circles
- Key improvement areas to inspect and adapt

Shortcut 24: Risk Takers and Mistake Makers

- The importance of establishing a failsafe environment to ensure open and honest feedback sessions
- Common fears that burden teams
- A selection of measures to help teams overcome these fears