WITH 40+ SIMPLE AND POWERFUL EXPERIMENTS

ZOMBIE SCRUM SURVIVAL GUIDE

A JOURNEY TO RECOVERY

CHRISTIAAN VERWIJS
JOHANNES SCHARTAU
BARRY OVEREEM

Forewords by DAVE WEST & HENRI LIPMANOWICZ

The Professional Scrum Series Scrum.org

Contents

Experiment: Diagnose Your Team Together

Throughout this book, you'll find many experiments and interventions that you can do with your team. They are all designed to help create transparency around what is happening, to allow inspection and encourage adaptation. Every experiment follows a similar pattern. We start with the purpose. Then we explain the steps and give direction on what to watch out for.

This first experiment is all about creating transparency and starting a conversation around Zombie Scrum (see Figure 3.6). This is a critical first step towards recovery and to confront the truth that work is needed. This experiment helps you progress on the first three steps of the First Aid Kit (Chapter 2): take responsibility, assess the situation, and create awareness.

This experiment is based on the Liberating Structure "What, So What, Now What?"[1] It is a good way to build confidence, celebrate small successes, and build the muscle to get through the hard stuff.

Skill/Impact Ratio

Skill	★☆☆☆☆	No skill is required for filling in a survey and inspecting the results together with your team.
Impact on survival	★★★★★	This experiment creates transparency around what is going on in your team (and around it) in terms of Zombie Scrum. It's a crucial first step on your way to recovery.

Figure 3.6 Team diagnoses in progress

1. Lipmanowicz, H., and K. McCandless. 2014. *The Surprising Power of Liberating Structures: Simple Rules to Unleash a Culture of Innovation*. Liberating Structures Press. ASN: 978-0615975306.

Steps

The following steps help you do this experiment:

1. Go to **survey.zombiescrum.org** and fill out the extensive free survey for your Scrum Team. Invite others from your team to join your "sample" as instructed. To protect others' privacy and avoid abuse of the survey, scores from individual members are only shown to each survey taker.
2. When you've completed the survey, you'll receive a detailed report (see Figure 3.7). The report will be updated every time someone joins the sample. In the report, you'll find results for the four symptoms of Zombie Scrum, as well as a more detailed breakdown. The report also gives feedback and recommendations based on the results.
3. When everyone has participated, schedule a one-hour workshop to inspect the results together. We recommend doing this with only the Scrum Team: the Product Owner, the Scrum Master, and the Development Team.
4. Prepare for the workshop. You can print the report and hand out copies, put prints on the walls, or simply put up the profile on a screen.
5. Start the workshop by reiterating the purpose clearly and emphasizing what will happen with the outcomes (and what won't). Make sure to emphasize that improvement is always a gradual, incremental, and often messy process and that this workshop is a step in that process.
6. Invite everyone to inspect the results silently and note down observations. Ask: "What do you notice in the results?" Encourage people to stick to the facts, and avoid jumping to conclusions, for the first round. After a few minutes, ask people to share their observations in pairs for another couple of minutes and notice similarities and differences. If you have eight or more people, ask pairs to join another pair and take a few minutes to share observations and notice patterns. Ask the small groups to share their most important insights with the whole group, and capture them in a way that remains visible to everyone present.
7. Following the pattern outlined in the previous step, repeat twice more with different questions. For round two, ask people "So, what does this mean for our work as a team?" For round three, ask people "Where do we

have the freedom and autonomy to improve as a team? What are small, first steps we can commit to?" Make sure to keep capturing the most salient outcomes.

8. Put the most important actionable improvement on the Sprint Backlog for the next Sprint. Involve others as needed to keep making progress.

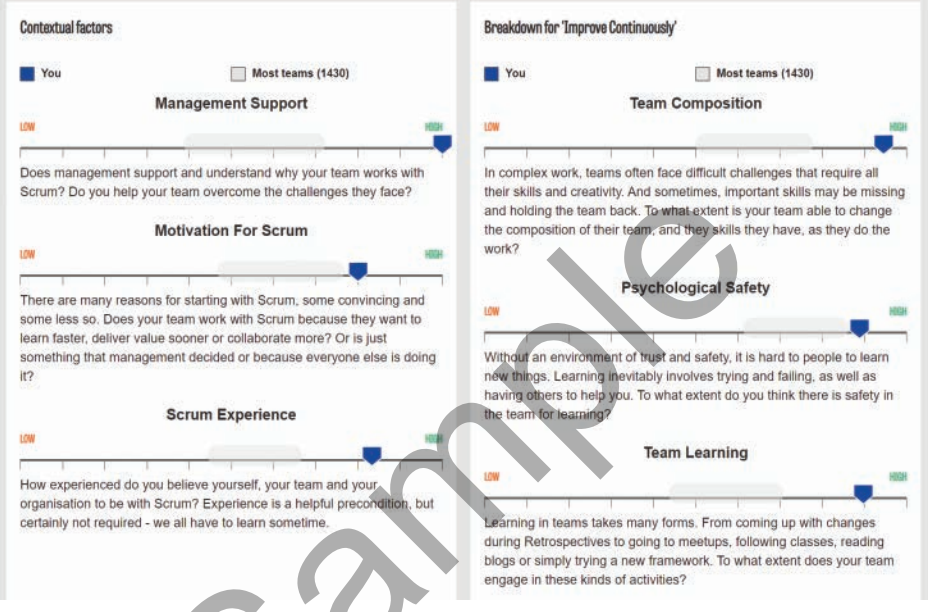

Figure 3.7 Part of the report you'll receive after completing the Zombie Scrum Survey

Our Findings

- It can be tempting to identify dozens of potential improvements and end up doing nothing at all. Instead, keep a strong focus on improving one thing first before moving onto something else. If that improvement is too big to commit to doing it in a single Sprint, make it smaller.
- When you ask people to participate in this survey, you're asking them to trust you with their honest answers. Be deeply respectful of that. Don't spread reports to people outside of the team or forward them to

management unless you have clear and unambiguous approval from everyone involved.

- Don't use the report to compare teams. Doing so will erode trust much faster than you can rebuild it.

Increase Cross-Functionality with a Skill Matrix

Is your team experiencing bottlenecks because only one person is capable of testing work? Is a developer on your team struggling to implement something that is blocking everyone else until she is done? Do team members start work

on unrelated and low-value tasks simply because they have nothing else to do? These symptoms arise when teams are not cross-functional enough, causing work to pile up for some people and creating delays for others.

The Scrum Framework is built on cross-functional teams because they are better able to overcome the unpredictable challenges that arise when working on complex problems. Your team is cross-functional enough when items flow smoothly through your workflow. Cross-functionality does not mean that everyone can perform any kind of task or that you must have at least two experts for every kind of skill on your team. Often, just having another person who has a particular skill, even when they are slower and less experienced at it, already improves flow enough to prevent most problems.

This experiment offers your team practical strategies to help them improve their cross-functionality (see Figure 8.4).

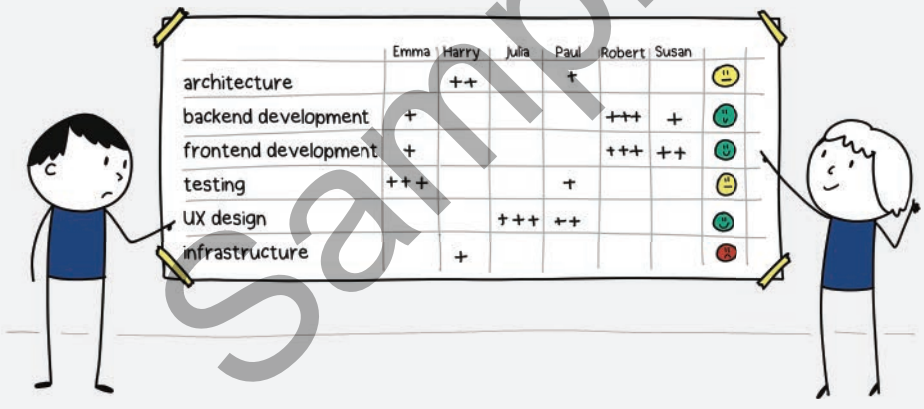

Figure 8.4 Increase cross-functionality with a skill matrix.

Effort/Impact Ratio

Effort	★★★☆☆	This experiment aims at one of the toughest causes of Zombie Scrum. You may have to deal with resignation and cynicism.
Impact on survival	★★★★★	Finding ways to distribute skills in your team not only improves flow, it is also good for morale.

Steps

To try this experiment, do the following:

1. With your team, map the skills you need during a typical Sprint. Together, create a matrix on a flip chart where you plot the members of your team against the skills you identified. Invite people to decide for themselves what skills they possess and to self-rate their proficiency with it using plus signs (+, ++, and +++).
2. When you're done with the matrix, ask "What do you notice about how the skills on our team are distributed? What is immediately obvious?" Invite people to reflect on this question individually for two minutes, then for a few minutes in pairs. With the whole group, capture important patterns on a flip chart.
3. Ask "What does this mean for our work as a team? Where should we focus our improvements?" Let people reflect on this question individually, then in pairs for a few minutes, and then capture the biggest insights on the flip.
4. Ask "Where should we start improving? What first step is possible for us without needing approval from others or resources we don't have?" Let people reflect on this individually, then in pairs for a few minutes, and then capture the biggest insights on the flip chart. Use the strategies as described in the next section as inspiration when people struggle to see possibilities.
5. Keep the skill matrix in your team room and update it frequently. You can tie it to flow-based metrics such as throughput and cycle time, which should improve over time as cross-functionality increases. See the experiment "Limit Your Work in Progress" to learn how to do this.

There are many strategies for improving cross-functionality on your team.

- You can add people to your team who already have skills that you need. Although a seemingly obvious solution, adding skilled people isn't always possible. It's also doubtful how structural this solution really is, as it can cause "Skill Whac-A-Mole," where other skills then become bottlenecks and you have to add even more specialized people. Instead of maintaining

high degrees of skill specialization, it's often more effective to distribute skills.

- You can automate tasks that require scarce skills. For example, creating a backup of a database or deploying a release are critical tasks that are often performed by database specialists and release engineers. When you automate these tasks, you improve not only the speed of the activity, but also how frequently these tasks can be performed, while also removing the constraint.
- You can purposefully limit your team's work in progress, putting constraints on how much new work can be started, to encourage cross-functionality. Instead of starting a new Product Backlog item, because there isn't anything else to do, ask "How can I help others complete their current work?" or "How can others help me complete this work?" The Daily Scrum is a natural opportunity to offer and request help.
- You can encourage people to pair on tasks that only a few people can perform. When you pair experienced and inexperienced people, the less experienced people develop new skills, and both people find better ways to support each other. For example, pairing developers who typically work on the front end with developers who work on the back end makes it easier for them to support each other when bottlenecks occur.
- You can use approaches such as "Specification by Example"[5] to allow customers, developers, and testers to work together to develop automated test cases. In a similar vein, front-end frameworks (e.g., Bootstrap, Material, or Meteor) can make it easier for designers and developers to work together with a common design language for elements.
- You can organize skill workshops where people who are skilled at a particular task demonstrate how they perform it and help others perform it.

Our Findings

- When Scrum Teams have been affected by Zombie Scrum for a long time, they may have come to believe that nothing ever changes. You may even face understandable cynicism. If this is the case, start with the smallest

5. Adzic, G. 2011. *Specification by Example: How Successful Teams Deliver the Right Software*. Manning Publications. ISBN: 1617290084.

possible improvements to show people that change is possible and worth the time spent making it happen.

- When the skills of team members are narrowly specialized, they may struggle to see how broadening their skills will benefit the team. They may also fear losing their uniquely visible contribution to the team. Make an effort to celebrate the successes of the team to emphasize the collective outcomes over individual contributions.

Share an Impediment Newsletter throughout the Organization

The impediments that make it hard for Scrum Teams to work empirically often involve people across the organization. Helping these people understand the impediments and the problems they cause creates awareness that enables double-loop learning, which can lead to systemic improvements.

Effort/Impact Ratio

Effort		This experiment calls for nothing but courage and a dash of tact.
Impact on survival		Although painful, this experiment is a great way to create urgency around the biggest problems.

Steps

To try this experiment, do the following:

1. With your Scrum Team, ask everyone to silently write down impediments they see that are making it hard for them to build what stakeholders need or ship fast(er), or both. What skills are missing? Where is protocol getting in the way? Which people do they need, but don't have access to? After a few minutes, invite people to pair up to share and build on their individual ideas. Together, share all impediments and pick the three to five impediments that are most impactful (e.g., with dot-voting).
2. For the biggest impediments, ask "What is lost because of this? What would we and our stakeholders gain when this impediment is removed?" Capture the consequences for the various impediments.
3. For the biggest impediments, ask "Where do we need help? What would help look like?" Collect the requests for help for the various impediments.
4. Compile the biggest impediments, including their consequences and requests for help, in a format that you can easily distribute to everyone who has a stake in your work. It could be a mailing, a paper newsletter, a

blog post on your intranet, or a poster that you put in a heavy-traffic corridor. Include the purpose of your team and how to contact you. You can also include the accomplishments of your team, of course.

Our Findings

- Make sure to include (higher) management and consider informing them up front. Also, they will probably appreciate a shorter, more concise version of the newsletter.
- Transparency can be painful. Be honest but tactful in your messaging, and don't blame others or be negative. State what is happening and make clear requests for help.
- If you are planning to do this experiment frequently, make sure to include the accomplishments of your team as well. What is going well? What has changed since the previous newsletter? And most important: from whom did you receive (unexpected) help?

Make the Cost of Low Autonomy Transparent with Permission Tokens

The autonomy of teams decreases as their dependencies on external people increase. Some dependencies are explicit, such as when a Scrum Team needs someone outside the team to do something for them. Other dependencies are more implicit. Having to ask for permission or approval from someone outside the team in order to proceed is a good example. This experiment is about making transparent where and how often permission is required (see Figure 12.1).

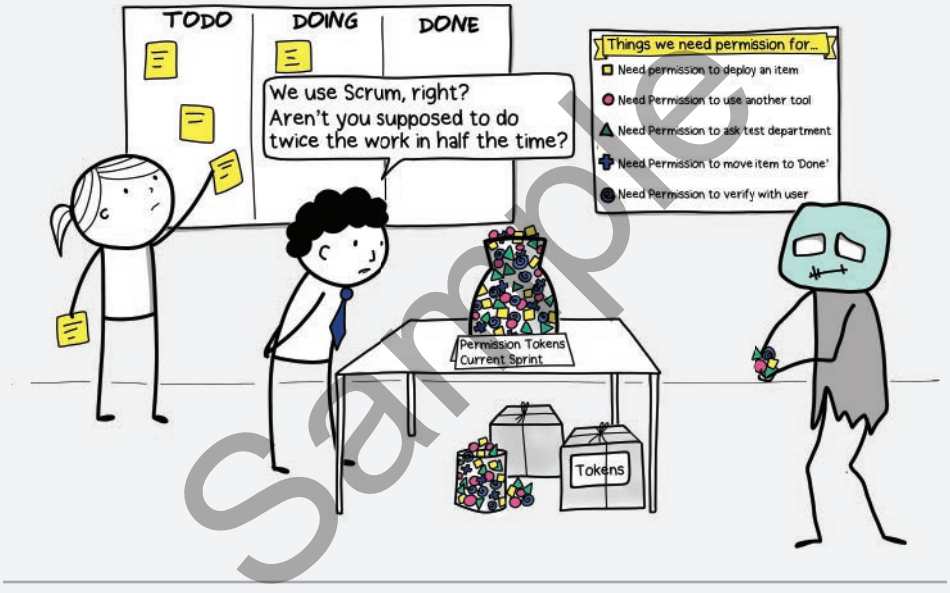

Figure 12.1 Without considering all the things that constrain Scrum Teams, it's easy to expect miracles from them.

Effort/Impact Ratio

Effort	★★☆☆☆	This experiment requires only a jar, some tokens, and a few minutes during your Sprint Review.
Impact on survival	★★★★☆	Even in the most zombified environments, regaining some sense of control makes people sigh with relief.

Steps

To try this experiment, do the following:

1. Find an empty jar, or another container, and place it in the team room. Somewhere near the Sprint Backlog is the best spot.
2. Give everyone on your team a bunch of permission tokens. You can use marbles, LEGO bricks, magnets, or stickies. Use different colors for the various permission categories. For example, the permission to release something, to move an item to another column on your Scrum Board, or to change your tools or environment. We recommend a limit of five categories to keep things simple.
3. During the Sprint, put an approval token in the jar every time someone on the Scrum Team has to ask permission from someone outside the team. For example, put a token in the jar when an external architect needs to approve that an item is done. Or when the Product Owner has to vet an item with an external manager. Put a token in the jar when you need permission from office management to purchase stickies. And put a token in the jar when you need a configuration to be changed by an external administrator. Aside from requests for permission, also add a token every time you need someone outside the team to perform a specific action as well.
4. During the Sprint Review, and with stakeholders present, share the number of tokens in the jar. Ask: "How does this affect our ability to quickly adapt in the moment and do what is the most valuable? Where can we simplify things?" Invite people to first consider this question for themselves and in silence, then in pairs for two minutes, and then paired with another pair for four more minutes. Capture the most salient improvements with the whole group. The Sprint Retrospective is a great opportunity for digging into potential improvements.

Our Findings

- For another perspective, you can use different colors for everyone on your team. This allows you to identify who is most often in need of permission.

- If you want to focus on the amount of organizational bureaucracy, don't add permission tokens for requests from direct stakeholders such as customers, users, or people who otherwise invest significant money or time in your product.
- The experiment "Break the Rules!" elsewhere in this chapter is great to test where asking for permission matters, and where it just gets in the way of doing the right thing.

Find Actions That Boost Both Integration and Autonomy

Organizations with self-managing Scrum Teams face the difficult challenge of balancing their autonomy while keeping their work integrated with the rest of the organization. Because both of these aspects are equally desirable, and we can't simply make an either-or decision, we are faced with what is called a "wicked question." Instead of letting the pendulum swing entirely to one side, this experiment is about finding ways of supporting both sides. With this approach, you help groups move from "either-or" to "yes-and" thinking. This experiment and its corresponding worksheet (see Figure 12.2) are based on the Liberating Structure "Integrated~Autonomy."[1]

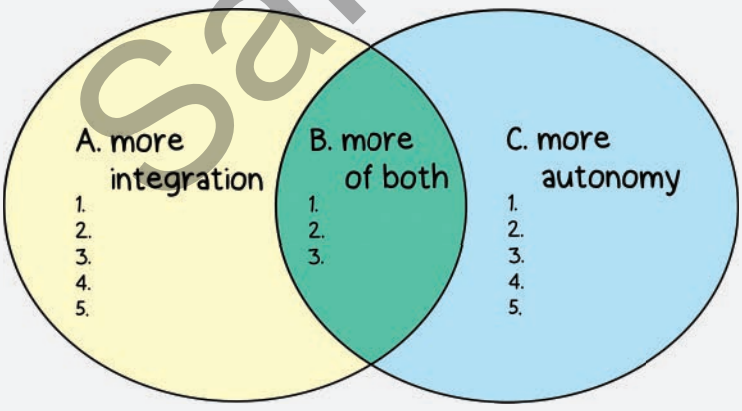

Figure 12.2 A simple worksheet for Integrated~Autonomy[2]

1. Lipmanowicz, H., and K. McCandless. 2014. *The Surprising Power of Liberating Structures: Simple Rules to Unleash a Culture of Innovation*. Liberating Structures Press. ASN: 978-0615975306.
2. Source: Lipmanowicz and McCandless, *The Surprising Power of Liberating Structures*.

Effort/Impact Ratio

Effort		This experiment greatly benefits from tight facilitation and asking powerful questions to help the group move out of deadlocks.
Impact on survival		As people start seeing that autonomy and integration are not opposed, more of both will be possible.

Steps

To try this experiment, do the following:

1. Invite people who have a stake in either increasing the autonomy of Scrum Teams or keeping them integrated with work done elsewhere. This includes the Scrum Teams themselves, departments they depend on (and vice versa), and management.
2. Begin by helping people make tensions between autonomy and integration tangible. Ask "For the Scrum Teams, where in their work is there tension between the desire for autonomy and the desire for integration?" Start with a minute of silent thinking (one minute), then invite people to share their ideas in pairs (two minutes). Capture salient examples from the whole group (five minutes). For example, there can be tension between the autonomy that Scrum Teams have over their Sprint Backlog and the need to be able to pick up urgent issues from people outside the team that emerge during a Sprint. There can be tension between the autonomy of a Product Owner to order the Product Backlog and keeping that ordering aligned with corporate strategy. Or between allowing Scrum Teams to pick their own tools and having mandated tools that are safe for corporate environments.
3. The next step is to explore actions that promote integration. For this step, the participants work with the Integrated~Autonomy worksheet shown in Figure 12.2. It shows three columns with space for writing down ideas that lead to either more integration (A), more autonomy (C), or both (B). The group will focus on column A first. Ask "What actions boost integration of the Scrum Teams' activities with what is happening

elsewhere?" Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the left side of the worksheet (ten minutes).

4. As a follow-up, explore actions that promote autonomy. Ask "What actions boost the autonomy of Scrum Teams?" Capture them in the right column of the worksheet. Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the right side of the worksheet (ten minutes).
5. Now that you have actions that each address one side of the wicked question, help the group move into yes-and thinking. Ask "Which actions boost both integration and autonomy?" Capture them on the worksheet in the middle. Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the middle of the worksheet (ten minutes).
6. Now that people have experience identifying actions that serve both sides, investigate earlier actions to see if they can be shifted to the middle. Ask "Which actions on the left or the right of the worksheet can be creatively modified to boost both integration and autonomy?" Start with a minute of silent thinking (one minute), then invite people to share their ideas in groups of four (five minutes). Capture the most salient actions from the small groups on the middle of the worksheet (ten minutes).
7. Order actions by their ability to promote both integration and autonomy and identify 15% Solutions for the most impactful ones (see Chapter 10).

Our Findings

- Coming up with specific and tangible actions can be difficult. Keep asking "How would you do that for us?" or "What would that look like here?" in order to move groups beyond abstract ideas and platitudes (such as "more communication").
- If you have a large group, you can make each group of four responsible for one of the actions you identified during step 2. Let them fill in the entire worksheet in their small group from the perspective of that action.

- You can replace the sides—integration and autonomy—with other wicked challenges. For example, there is also tension between responding to change as quickly as possible and preventing huge mistakes. Or the tension between standardization on the one hand and customization on the other. Work with whatever wicked challenge makes the most sense!