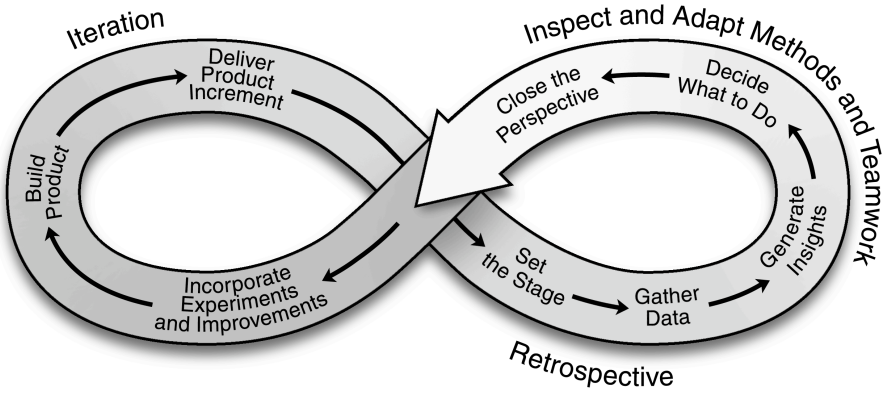
Agile Retrospectives: Making Good Teams Great!

Esther Derby and Diana Larsen
Agile 2007
Washington, D.C.



Set the Stage

Lay the groundwork for the session by reviewing the goal and agenda. Create an environment for participation by checking in and establishing working agreements.

Gather Data

Review objective and subjective information to create a shared picture. Bring in each person's perspective. When the group sees the iteration from many points of view, they'll have greater insight.

Generate Insights

Step back and look at the picture the team created. Use activities that help people think together to delve beneath the surface.

Decide What to Do

Prioritize the team's insights and choose a few improvements or experiments that will make a difference for the team.

Close the Retrospective

Summarize how the team will follow up on plans and commitments. Thank team members for their hard work. Conduct a little retrospective on the retrospective, so you can improve too.

From *Agile Retrospectives: Making Good Teams Great!* by Esther Derby and Diana Larsen.
Published by Pragmatic Bookshelf (2006).
Available at http://pragmaticprogrammer.com/title/dlret/index.html

SCENARIO # 1

Team Size:	5
Length of project/iteration:	1 week
End of Project: ☐	Release: ☐ Iteration: ☑
Previous retrospectives:	Yes, after every iteration. This is their 6th retrospective.

Background:

The team is using iterative incremental development and holding a short retrospective every Friday afternoon. The team coach has been leading the retrospectives. So far he's been asking the team What they should Start/Stop/Continue, looking for ways to improve their development methods. The last couple of retrospectives the team hasn't come up with many new ideas. Further, the team members seem bored. At the last retrospective two team members said they thought retrospectives were a waste of time.

SCENARIO # 2

Team Size:	20	
Length of project/iteration:	30 days	
End of Project: ☐	Release: ☐	Iteration: ☑
Previous retrospectives:	No	

Background:

The team hasn't done a retrospective before, but this iteration was such a disaster that the VP decreed that they must hold a retrospective to learn what went wrong. The team is worried that the VP is looking for someone to blame, and that the coach will buckle and blame the team. After talking to the VP, you understand that this isn't his motive, he genuinely wants to learn from the failure, and understand what he –as well as the rest of the team-- can do differently next time.

SCENARIO # 3

Team Size:	10	
Length of project/iteration:	30 days	
End of Project: ☐	Release: ☐	Iteration: ☑
Previous retrospectives:	No	

Background:

This team is new to iterative incremental development and this is their first retrospective. They're excited that they've finished a feature and showed it to the customer. None-the-less, not everything went smoothly on their first iteration, and they want to continue to improve their engineering practices. They particularly struggled with stepping on each others code during the iteration.

SCENARIO # 4

Team Size:	8
Length of project/iteration:	2 weeks
End of Project: ☐	Release: ☑ Iteration: ☐
Previous retrospectives:	Yes, 1-hour iteration retrospective after each 2-week iteration. Retros have been facilitated by the team coach.

Background:

The team is proud that they met the release goal and want to keep up momentum for the next release. Things worked pretty well within the team, but there was friction with the operations support area both in getting servers ordered and in accepting turn over. The team is also struggling with customer defect reports.

SCENARIO # 5

Team Size:	8	
Length of project/iteration:	30 days	
End of Project: ☐	Release: ☑	Iteration: ☐
Previous retrospectives:	No.	
Background:		

The team delivered less functionality than planned, but did meet the date. Because the code was late, the test team had less time to do their work and dropped tests in order to make the date. The test manager recommended pulling two features, so those features were dropped from the release at the last minute.

Extracted from:

Agile Retrospectives

Making Good Teams Great

Chapter 3

Leading Retrospectives

This chapter is about the role and skills of a retrospective leader. You don't need to be a professional facilitator to lead an iteration retrospective, but you do need basic facilitation skills. To learn the skills, you need to understand the role, practice, and seek feedback.

As a retrospective facilitator you may follow the content, but your primary responsibility is the process. When facilitators talk about process, they aren't talking about a heavyweight methodology. *Process* means managing activities, managing group dynamics, and managing time [Sch94]. Retrospective leaders focus on the process and structure of the retrospective. They attend to the needs and dynamics of the group and help the group reach a goal. Retrospective leaders remain neutral in discussions, even when they have strong opinions.

When the content involves your own team, it's easy to get caught up in the discussion. It's tempting to jump into an engaging conversation, especially when you care about the topic. But, if you're immersed in the content, you can't pay full attention to the process. Wait a beat to determine whether your thoughts are necessary. Most often, your team will do nicely without your input. The risk of giving input is that when the leader jumps in too often, it quashes group discussion.

Participants, on the other hand, focus on the content, discuss, sometimes disagree (though not disagreeably), and make decisions. Participants aim toward a goal and manage their own thoughts, feelings, and responses so they contribute positively to the conversations and outcome.

TIP: When to Offer Content Expertise

You may have important content to offer that no one else in the group has. When that happens, tell the team you are leaving the retrospective leader role temporarily to contribute to the discussion. Hand your marker to another team member to symbolize that you are not in the facilitator role while you participate. (Just make sure you get the marker—and your role—back.)

3.1 Managing Activities

Every retrospective design includes activities—such as creating working agreements, building a timeline, brainstorming, and prioritizing—to help the team think together. You'll need to introduce each activity, monitor the room during the activity, and debrief the activity when it's done.

Most people want to know something about the purpose of an activity before they begin. Give a broad sense of the territory the team will explore without revealing the details of what will happen or specifying what the team will learn.

TIP: Introducing Activities

The first time you use an activity, write a script so that you remember what to say and don't garble the instructions or leave something out.

Once you have your script, practice saying it aloud. Saying the words is different reading them or thinking them. As you hear yourself give the instruction, you'll notice where you stumble and where even you can't follow the instructions. Then you can refine your script and practice again.

You may not follow the script in the end, but preparing and practicing will help you describe the activity clearly and concisely.

Here's an introduction for an activity to re-create the timeline of a release: "To understand our iteration we need to tell the whole story from everyone's perspective. We'll create a timeline that shows events

that happened during the project. After we have a timeline as complete as it can be for now, we'll look for interesting patterns and explore puzzles."

This tells you about the territory of "understand our iteration," and lists the steps at a high level: "create the timeline," "look for interesting patterns," and "explore puzzles." It doesn't tell your team exactly what the outcome will be. That's for the team to create.

Most people (even really smart people) can't absorb detailed instructions for a multipart activity. Give the details for each part, just in time. For the timeline the details of the first steps are as follows: "Let's get into groups of two or three. In your group, brainstorm all the events that took place during the release. An event doesn't have to be a milestone—it can be anything that happened on the project." After giving the instructions, ask for questions about the task. Pause. Count to ten. Someone will have a question—wait for it.

As the retrospective leader, you have two tasks during an activity: be available to answer questions about the activity, and monitor the room.

While the group is working on an activity, listen to the noise level. Lots of conversation is an indication of good energy. It's also a clue that people are done with a quiet activity or need more time for a discussion activity. For an activity that involves writing or individual work, conversational buzz indicates that people are done and have started talking to their neighbor. If it sounds like there's still lively conversation at the end of a discussion activity, check to see whether people need more time. Of course, the sound of lively conversation may mean that people have finished the task and are talking about the latest movie.

Debrief every activity. A debrief helps your team examine their experience and extract insights. They'll make conscious connections and form new ideas. Debriefing each activity builds toward the insights and decisions of the retrospective.

So, it's important to debrief. Now how do you do it?

Here's a simple, four-step method to debrief almost any activity [Sta97]:

1. Start by asking for observable events and sensory input. "What did you see and hear?"
2. Ask how people responded to those events and inputs. "What surprised you? Where were you challenged?"

3. Ask for insights and analysis with questions, like "What insight do you have about this?" followed by "What does this tell you about our project?" These questions help people formulate their ideas and connect the activity to the project.

4. After you've established the link between the activity and the project, complete the learning cycle by asking group members how they will apply their insights: "What's one thing you might do differently?"

Notice anything familiar about this? If follows the same flow as the retrospective structure (gather data, both facts and feelings; generate insights; and decide what to do).

There are lots of other ways to debrief (see Appendix B, on page 152 for additional ideas). This is a good place to start.

For a five- to twenty-minute activity, spend 50–100% as much time on debriefing as on the activity. So for a ten-minute activity, allow five to ten minutes for debriefing.

3.2 Managing Group Dynamics

Most of the time managing group dynamics in a retrospective means managing participation: making sure people who have something to say have the chance and making sure people who have a lot to say don't dominate. Watch out for people who are talking more (or much less) than others. Make an opening for the quieter team members by asking to hear other opinions. Notice when someone looks as though he or she was about to speak but was cut off. Ask whether he or she has something to say. Create an opportunity without putting people on the spot or demanding an answer [Der03a].

To draw out quieter people, try saying something like "We haven't heard Leigh and Venkat yet. What would you add?" Be willing to accept a pass.

If someone just won't shut up, be direct (in private). If you've observed the pattern, talk to the person before the retrospective. Describe your observations, and describe the impact on the team—other people have stopped participating. Ask him or her to hold back. If the private conversation doesn't work, be direct in the retrospective. When one team member is first to speak on every question, hold up a hand and say, "We've heard from you on every question; let's hear from some

other people." Keep your tone neutral. An emphatic delivery—"We've HEARD from YOU on EVERY question"—conveys blame and won't help the retrospective.

Managers won't be in every retrospective, but when they are, they are particularly prone to dominating the conversation. It's not always their fault—if team members hold back when a manager is in the room (for whatever reason), the manager tends to fill the dead air. Meet with managers before the retrospective. Coach them on appropriate participation. Ask them to let others talk first, acknowledge the contributions others make, and be careful how they disagree. "I see it differently" preserves participation. Statements like "You're wrong," "You just don't understand," "You're not listening to me," and "I disagree" quash participation or lead to confrontation. Neither one is good.

Here's how one retrospective leader handled a talkative manager: Rajiv, a project manager was a high-energy, verbal guy. And he was excited about the project. Jess met with him before the retrospective to discuss participation. Rajiv worried he'd forget to wait for others to speak first. Jess and Rajiv agreed on a signal: if he spoke out of turn, Jess would walk over and stand next to him. They never used the signal. Just knowing it was there was enough to help Rajiv wait.

TIP: Strategies for Helping Your Team Move Forward

Sometimes teams become stuck. When that happens, you have options as a retrospective leader.

You can help restore their creative juices by asking questions such as these:

- What have we tried before? What happened? What would you like to happen differently?
- If we had that, what would we gain?
- Have you ever tried this a different way? What happened?

You can ask for more opinions, especially from people who have been thinking more than talking.

You can suggest additional research before committing to a solution.

You can take off the retrospective leader hat and offer content knowledge from personal experience.

You could tell the team what to do, but only if you want to cheat their learning.

After managing participation, the next most common issues are violating working agreements and blaming. Both have negative effects, so you don't want to let them pass unnoticed.

Sooner or later, a team member will violate a working agreement. Humans have good intentions but fall back into old patterns. When they do, remind your team of their working agreements. If you allow violations to continue without comment, team members get the message that working agreements are optional. Optional working agreements have no value. It's everyone's job to monitor working agreements.

Blame starts a downward spiral of defensiveness and counterblame that will torpedo a retrospective. Listen for "you" language ("You broke the build!") and labeling statements ("You're immature!"). Both signal blame. Blame hurts the retrospective by distracting attention from real problems.

Encourage "I" language. "I" language centers on the speaker's observation and experience, rather than labeling the other person. When you hear blame or personal criticism, intervene and redirect the discussion to the content.

Here's how one retrospective leader handled a blame: during the platform expansion retrospective, one team member blamed another for breaking the build. "We'd have met our target if it weren't for you!"

"Hold on!" the retrospective leader said, "can you say that using 'I'? language?" The team member thought for a while and then said, "I am angry that we missed our target because we had so much trouble fixing the build." Then the team was able to look at bigger issues with the build without blaming one individual.

Describe what you've seen and heard: "I'm hearing labels and 'you' language." Describing the behavior causes people to pause and consider what they're doing.

Group dynamics include team member interactions and emotions. You aren't responsible for other people's emotions, but as retrospective leader, you are responsible for keeping the session productive. And that

means you need to be prepared to handle emotional interactions and situations.

Most interactions and emotions help the group move forward. Some don't. Here are some challenging group dynamics and interactions to watch out for—and what to do about them. With any luck you won't encounter all of these in one retrospective ;-). If outbursts are the rule on your team, something else is happening. Retrospectives can't solve every problem; if the issue is deeper than normal team friction, contact your HR representative for resources and guidance.

When people have been bottling up their emotions, they come out in funny ways: people cry, shout, stomp out of the room, laugh hysterically, or clown when the topic is serious.

Before you jump in to fix things, notice your own response. It's easy to focus on comforting one person and lose track of the goal and the needs of the team. In a retrospective, your primary responsibility is to the interactions of the team as a whole, not to individuals. That doesn't mean ignoring what's going on with individual emotions; it means dealing with emotions in a way that is helpful and respectful to the team and the individual.

Here are some strategies that have worked for us and can work for you. Having a mental picture of how you'll respond gives you more options in the moment. So, think of the outburst that scares you the most, and mentally rehearse using one of these strategies. Outbursts are unsettling, but they don't have to derail the process. If you think you could never do something like this, remember that one of the Retrospective Goddesses started out as a programmer.

Tears Offer a box of tissues. When the person is able to speak, ask, "What is happening for you? Can you share it with the group?" Pause. Given time, the person often shares something heartfelt (and usually relevant) about the topic under discussion.

Shouting In most places, when someone starts shouting, the rest of the people in the room stop participating. And that makes it unproductive for everyone. Intervene immediately. Hold up one hand as a stop sign, and say calmly but forcefully, "Hold it." Then say, "I want to hear what you have to say, and I can't when you're shouting. Can you tell us why without shouting?" Don't be surprised if the person responds, "I'm not shouting!" When someone is upset or excited, he or she may

not be aware of the rising vocal volume. There's no need to say "Yes, you are." Calling attention to the yelling is usually enough to stop it.

If your team member continues to blame or yell, call a break, and talk to the person privately. Let him or her know how the behavior is affecting the group. Ask for agreement to express emotion in a nonthreatening way. If the person is unwilling, ask (don't tell) him or her to leave and return when he or she has more self-control.

Stomping Out When a team member stomps out, let him or her go. Ask the team, "What just happened?" They will have an idea. Ask whether it is possible to continue without the person who left. Most of the time, they'll say they can continue, though they may need to talk about the departure.

If this happens more than once, another issue is at play. Talk to the individual outside the retrospective.

Inappropriate Laughter and Clowning It is great to have fun in a retrospective. And people may use laughter and humor to deflect from a sensitive topic. When the laughter has an edge or your team repeatedly avoids a topic, it's time to step in. Make an observation, and ask a question: "I've noticed that every time we get near this topic, someone tells a joke. What's happening?" They'll tell you and engage the topic.

Also watch for two other types of situations. They aren't outbursts, but they are worth noticing.

Uncharacteristic Silence When a team that has been voluble goes quiet, something is going on. Again, step in with an observation and a question: "It seems to me that the group is being awfully quiet. There was a lot of energy and conversation earlier. What's going on now?" Your team may just be tired and need a break. Or they may be unsure how to approach a topic. Once you ask the question, someone will figure out how to broach the topic, and the proverbial dam will burst.

Of course, the fact that a team goes quiet may not mean anything. They may be thinking, tired, or simply a quiet group. When the silence is sudden or out of character, it's a clue worth following.

Currents Beneath the Surface Fidgeting and intense side conversations may indicate something going on just beneath the surface. Again, ask the group what is going on. They will tell you.

Here's how one retrospective leader handled a sudden disturbance in a retrospective: During an off-site release retrospective for a team build-

ing network infrastructure, Lindsey noticed the manager take a call on his cell phone, even though the working agreements prohibited calls. He left. When he came back in the room, he had a quiet side conversation with one person, then another. Laptops opened. Everyone was still trying to track the discussion in the room, but something was distracting them. Lindsey stopped the discussion and asked, "What's happening?" A team member explained that there was a crisis back at the office and the sales manager wanted them to come back and fix it. They wanted to stay in the session but were distracted by his request and perceived urgency for the customer. Lindsey and the team discussed their options: stop the retrospective and reschedule it, ignore the request, or try to do something from where they were. The team set a timebox for immediate problem solving in the room and then resumed the retrospective.

Lindsey didn't blame anyone for not following the agreements. In most cases, naming the behavior, commenting on it, and asking the group what's happening will diffuse the situation and shift the dynamic.

Whew! After all that, managing time will be easy!

3.3 Managing Time

Here's the rub: when you're leading a retrospective, you should respond to the needs of the group, *and* you need to pay attention to time and stay within the timebox all at once. It's a dilemma.

Bring a timepiece that will allow you to time activities. We sometimes lose track of time, so we'll often jot down the start time so we know when to end an activity. Or you can use a stopwatch to time activities.

If you're working with a group much larger than eight people, you'll need a way to cue people that it's time to move to another step. Use a bell, chime, or some other not-too-obnoxious sound when it's time to come together as a group, debrief, or provide additional instructions for an activity. Yelling over the group isn't effective and sends the wrong message. Whistling works to gain attention, but not always with the desired effect. Duck calls, cow sounds, and other animal sounds work in groups of less than ten (the sound doesn't carry in larger groups), but well... they don't add to your dignity (if you care about such things).

When the discussion still has energy yet the time you've planned has run out, ask the group what they want to do: "I'm concerned that if

we continue this discussion we won't meet our end goal. What do you want to do?" The group will refocus and move ahead, or they will tell you this conversation is more important than the original goal. Put the decision in the hands of the group.

Usually it's pretty clear. When it's not, look for a compromise such as timeboxing the discussion or agreeing to revisit the topic later (in the retrospective or afterward).

Be prepared to swap to a shorter activity if time is running short. You still have the responsibility to meet the goal of the retrospective—identify and plan for experiments and improvements.

3.4 Managing You

In addition to managing activities, group dynamics, and time, you need to manage you.

Staying aware of all these team and interpersonal dynamics may seem overwhelming. The key to managing group dynamics is not technique (although it helps to have strategies) but in understanding and managing your own emotional state and responses. If you aren't managing your own state, no technique or strategy will help. When emotions are high, your team needs someone to stay outside the turmoil. That someone is you, the retrospective leader.

If you feel your anxiety or tension rising, take a deep breath. Call a break if you need to do so. Your anxiety is a clue that you need to sort out what to do next to serve the group. Remember, you didn't cause the emotions in the room, and you don't have responsibility to make everything and everyone happy and nice.

During the break, take a moment to shake out your hands and feet to release tension and get your blood flowing again. Take three deep breaths. This may seem like superfluous advice; but when people are tense and anxious, it reduces blood flow to the brain... which reduces the ability to think clearly, which contributes to anxiety and tension. You see the picture. Oxygen to the brain is a good thing. It helps you think. Once your brain is oxygenated, ask yourself these questions:

- "What just happened?"
- "How much was inside me, and how much was outside me?"
- "How did the group get here?"

- "Where does the group need to go next?"
- "What are three options I have for next steps?"
- "What will I offer the group?"

These questions will help you re-center. And then you can use one of the strategies for managing group dynamics. As long as you have a strategy, you won't have to stand there frozen, not knowing what to do. Over time, your comfort in dealing with charged emotional situations will grow. Find a mentor whom you have seen manage emotions in groups. Work with your mentor to gain confidence and learn more options for handling emotional situations. And remember to breathe.

3.5 Taking Your Skills to the Next Level

If you enjoy helping groups think together, increase your skills as a facilitator and augment your toolbox. Consider deepening your skills in these areas:

- Working with activities. There's an art to developing, introducing, and debriefing activities and simulations to help people think and learn together. In addition to using activities in retrospectives, using activities and simulations is helpful if coaching, teaching, or training is part of your job.
- Helping groups reach decisions. There's a huge body of knowledge related to how people really make decisions (it's not entirely by logic, by the way). You can improve the quality of decision making in your group by knowing what decision process fits the situation and how to help the group converge on a decision.
- Understanding and managing group dynamics. Learning about people and people in groups is a lifelong study. Your skills in this area will help you build and nurture high-performing groups as well as run a darn good retrospective.
- Increasing self-awareness. Self-awareness is the foundation of personal effectiveness. You can't go wrong learning more about yourself and learning how you respond under stress. Gaining awareness of habitual patterns is the first step to being able to choose an appropriate response rather than simply reacting.
- Creating and using flip charts. Don't use any more of those scribbled flip charts that no one can read from more than a foot away!

If you work with groups, learning how to present information visually helps the group process information quickly and efficiently.

These skills apply in many situations, not just retrospectives. Your understanding of group process and your ability to help groups succeed will help you succeed, too.

Practice facilitating other kinds of meetings. If you belong to a volunteer group or some other organization outside of work, offer to facilitate a meeting or subcommittee. It's low risk and will give you experience. Practice in managing the dynamics of any meeting will pay off in managing the dynamics in a retrospective.

Observe other people who are effective at leading meetings and working with groups. Watch how they interact with people and how they respond when a session isn't going smoothly. You may not want to use someone else's exact words, but you can analyze what you see and adapt it to fit your own style.

Practice with feedback is the best way to learn facilitation skills [Der02]. Ask someone you trust (and who has some facilitation awareness) to observe as you facilitate. If you have a specific area you want to learn about, ask your observer to pay special attention to that aspect of your facilitation. Or you may ask your trusted observer to look for areas where he or she senses you have habits you aren't aware of.

For resources on increasing your facilitation skills, see Appendix D, on page 156.

You are probably an expert at what you do now. Facilitation draws on different skills than most of us develop working in software. Facilitation also requires a different perspective. It takes time and practice to feel comfortable with new skills. Give yourself time, manage your expectations, and find mentors. You'll inspect and adapt your facilitation, too.

Extracted from:

Agile Retrospectives

Making Good Teams Great

5.1 Activity: Timeline

Use this to gather data in a longer iteration, release, or project retrospective.

Purpose

Stimulate memories of what happened during the increment of work. Create a picture of the work from many perspectives. Examine assumptions about who did what when. See patterns or when energy levels changed. Use this for "just the facts" or facts and feelings.

Time Needed

Thirty to ninety minutes, depending on the size of the group and the length of the increment of work.

Description

Group members write cards to represent memorable, personally meaningful, or otherwise significant events during the iteration, release, or project and then post them in (roughly) chronological order. The retrospective leader supports the team to discuss the events to understand facts and feelings during the iteration, release, or project.

Steps

1. Set up the activity by saying "We'll fill in a timeline to create a fuller picture of this iteration/release/project. We want to see it from many perspectives."

2. Divide the team into small groups, with no more than five in a group. Keep people who worked closely with each other together (affinity groups). It's better to have two small groups representing one affinity than one big group.

   Hand out markers and index cards or sticky notes.

   Make sure each person has a marker. Although it sounds schoolmarmish, you do need to remind people to write legibly, so people can read the cards.

3. Describe the process.

Ask people to think back over the iteration/release/project and remember all the memorable, personally meaningful, or significant events and write them down, one per card or sticky note.

Remind the group that the point is to see many perspectives—so they don't arrive at a consensus of what was memorable, meaningful, or significant. If it was any of those to one person, that's enough.

Tell them they have ten minutes for this activity.

If you are color coding (see "Variations") explain what the colors mean and post a legend.

Remind people to write legibly.

4. Monitor the level of activity as people start talking about events and writing cards. If people haven't started writing cards after half the time has elapsed, remind them to start writing. When the groups have a stack of cards, invite people to start posting them (see Figure 5.1, on page 54).

5. When all the cards are posted, invite the team to walk by the timeline and see what others have posted. It's OK for people to add new cards at this point as they remember more events.

6. Allow a break or take lunch before analyzing the timeline.

Variations

We have collected several variations on the timeline activity. We use index cards, sticky notes, markers and dots in a number of ways to pull out both fact and feelings data. For example: For example:

Color Coding Feelings To gather both facts and feelings, use colors to represent emotional states. For example:

- Blue = sad, mad, bad
- Red = challenged, stalled
- Green = satisfied, successful, energetic
- Yellow = cautious, confused
- Purple = fun, surprise, humor
- Salmon = fatigued, stressed

Color Coding Events Use colors to represent types of events. For example:

- Yellow = technical or technology related
- Pink = people or team related
- Green = organization related

Color Coding Functions Use colors to represent functions. For example:

- Blue = developers
- Pink = customers
- Green = QA and testing
- Yellow = technical writers

Color Coding Themes If the team wants to focus on particular matters, use colors to identify events related to specific themes. For example:

- Yellow = team communication
- Blue = equipment usage
- Pink = relationships with customers
- Green = engineering practices

You can pick your own color scheme based on the cards and sticky notes available to you.

Functional Swim Lanes Draw rows lengthwise along the backdrop for the timeline (assuming you aren't planning to post cards directly on the wall—then use ribbon or tape to demarcate the rows). Make a row for each department or function. That group will place their cards only in that swim lane.

In/Out Swim Lanes Draw one line that divides the backdrop in half lengthwise. Use one half for cards for team events and the other half for participants who were involved in the project but weren't part of the core team.

On/Off Use some special shape to represent the people on the project—stars or people-shaped cutouts are good. Ask people to represent when they started on the project by posting a star/people cutout on

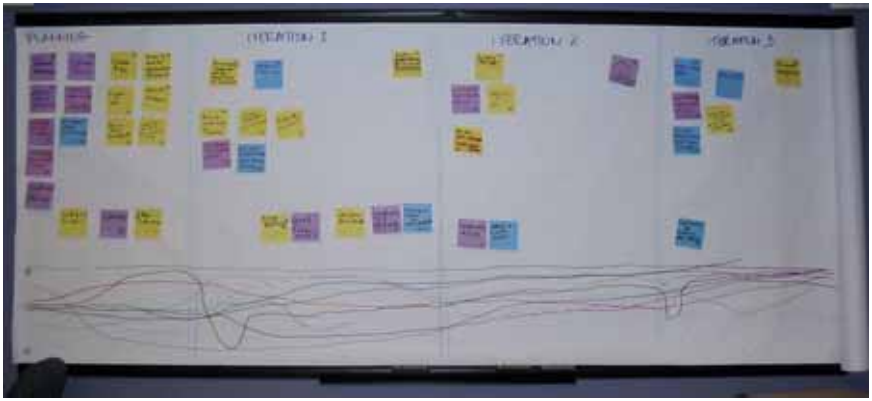


Figure 5.1: A timeline for a retrospective that looked at three iterations. The team was just starting retrospectives and wanted to look back further than just one iteration.

the timeline. Add a star or cutout for people who are no longer on the project or who aren't in the retrospective.

Materials and Preparation

Markers. Index cards or sticky notes. Drafting dots or some other movable tape that allows people to reposition event cards. Painter's tape or tacky stuff to paper the wall.

Backdrop. Cover a long wall with paper to serve as a backdrop. You can overlap flip chart pages or use roll paper. A stretch 6 feet long and 30 inches high is about right for a one-week iteration. For a longer project, you may need 30–60 feet long by 4–6 feet high.

Paper the wall before the retrospective starts.

(For a release or project, prep the timeline with a few time markers such as project milestones, months, or seasons.)

Example

A timeline can display many levels of data about the iteration, release, or project. It can be a simple, chronological listing of events on white index cards. It can also be an extravaganza of data including color-coded themes, cards arranged high or low for meaning, swim lanes for

different functional areas, dots to show positive and negative events, and a space at the bottom with a graph for the ongoing emotional ups and downs. It's easy to get carried away with the possible variations and ask teams to create a timeline with more data than they have time or mental energy to discuss.

When you have only an hour or so for the entire retrospective session, choose a timeline variation that will help to display just enough data. Include both facts and feelings, for sure, but no more than one kind of each. Consult the retrospective goal as a guide for what's most important this time. Keep it simple.

5.2 Activity: Triple Nickels

Use this to gather data or as part of the Decide What to Do phase in an iteration, release, or project retrospective.

Purpose

Generate ideas for actions or recommendations. Uncover important topics about the project history.

Time Needed

Thirty to sixty minutes, depending on the size of the group.

Description

Form small groups. In the groups, each person has five minutes to brainstorm and write down ideas individually. At the end of five minutes, each person passes the paper to the person on his or her right. That person has five minutes to write down ideas that build on the ideas already written on the paper. Repeat until the paper returns to the original writer.

Steps

1. Set up the activity by saying "In this activity, our goal is to generate as many ideas as we can about [topic]." Then explain the process (see the brief description earlier).
2. Divide the team into small groups, with no more than five in a group. Hand out paper for people to write on. Make sure each person has a pen or pencil. Remind people to write legibly so the next person can read the ideas.
3. Describe the process: In the first round, each person will have five minutes to write down ideas related to the topic. Aim for at least five ideas. In subsequent rounds, each person writes ideas that are sparked by the already written ideas or builds on them in some way.
4. Time the group. After five minutes, ring a chime and tell the group to pass the paper to the right.
5. Ask each person to read the ideas listed on the paper.
6. Debrief using these questions:

- What did you notice while you wrote ideas?
- What surprised you? What met your expectations? How?
- What is missing from these lists?
- What ideas and topics should we examine further?

Segue into the next activity where the team will use the ideas generated.

Materials and Preparation:

Paper. Pens or pencils.

Variations

If there are seven or fewer people in the group, don't divide into small groups; do the activity as whole group. Pass the paper only five times.

Examples

With a team of mostly reticent developers and one or two outspoken individuals, an activity like Triple Nickels can allow team members time to think privately yet participate in a process that develops whole-team understanding. It also prevents the few people who are comfortable talking in a group from dominating the discussion. In Triple Nickels, everyone gets the chance to contribute equally to developing the data set, and by the time the data is out, even the more reticent folks usually have something to say about what they wrote or read.

To help the five members of an internal business applications team gather data about their iteration, Aswaria, the retrospective leader, introduced the Triple Nickels activity. She divided the ten people on the team into two groups and passed out paper tablets and pens.

"I'll give you each five minutes to write down five important events that happened during our iteration. Record things you saw or heard during the past fifteen days. Write legibly; make sure someone else can read it."

At the end of five minutes, she said, "Now pass your papers to the right. Read what you get. You have five minutes to add detail to the items there or add new, related events."

The team kept passing the papers until each member received their original paper back to read. Some team members laughed at comments; others were shaking their heads. To maintain the theme of "fives," Aswaria debriefed with questions such as the following: "What five things stand out for you about what you've read?" "What five events produced the strongest reactions?" "What are the five most significant events?"

After they finished the discussion, she handed out sticky drafting dots and invited people to post the papers on an area of the wall that she had labeled "Iteration History."

5.7 Activity: Team Radar

Use this to gather data in an iteration, release, or project retrospective.

Purpose

Help the team gauge how well they are doing on a variety of measures, such as, engineering practices, team values, or other processes.

Time Needed

Fifteen to twenty minutes.

Description

Team members track individual and group ratings for specific factors about process or development practices they want to examine.

Steps

1. Introduce the activity by saying "We agreed on these [fill in the factors] as important to our work. Let's assess how well we are doing with them, using a scale of 0–10. Zero means not at all, and 10 means as much as possible."
2. Post the flip chart with the blank radar graph. Ask each team member to approach the chart and place a dot or some other mark that shows their rating for each factor.
3. Lead a short discussion about how the factors affect the work of the team. Consider asking questions such as the following:
   - Where do you see us following these [fill in factors]?
   - Where do you not see us following these [fill in the factors]?

   Use the short discussion as a segue to generating insights.
4. Save the completed flip chart graph. Run the activity again two or three iterations later. Compare the two charts as a progress measure.

Materials and Preparation

Flip chart or white board. Markers.

If you know ahead of time what the team will measure using the radar chart, draw the spokes and label them ahead of time (See Figure 5.5

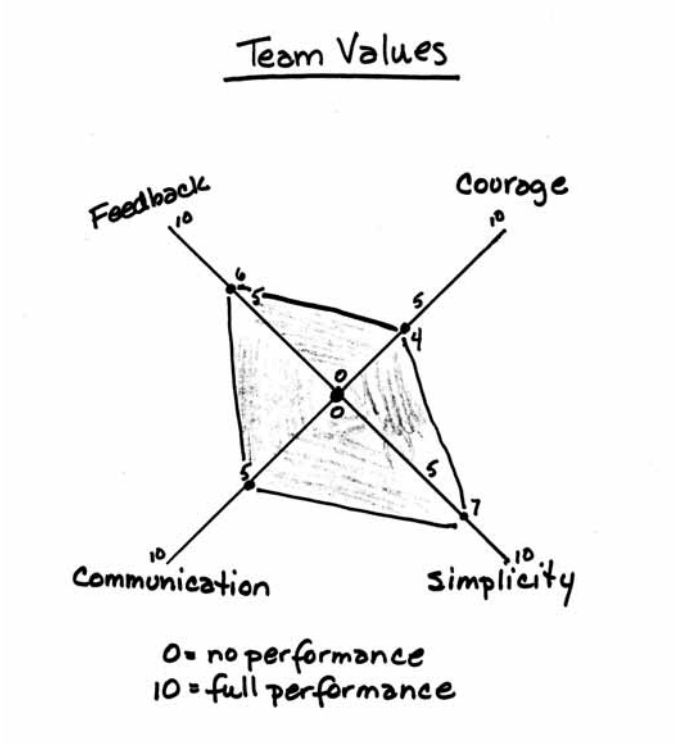


Figure 5.5: This team used the Group Average Radar to gauge how much they were following their team values.

). If the team will brainstorm the measures during the retrospective, draw the radar chart during the retrospective.

Variations

You can use this activity to measure many different factors, such as, engineering practices, team values, working agreements, methods, and so forth.

Group Average Radar This variation is an ongoing measure of progress on a particular measure. Use the radar chart but instead of collecting individual responses, calculate the group average for each measure.

Hand each team member a set of colored cards, one for each factor measured. Ask each person to rate each factor from 0–10 and hand the

card to you. Shuffle the cards (within colors) as you receive them so it's not clear which card came from a particular team member.

Recruit a team member to help with calculating averages. Post the averages on the radar arms. Connect the dots, and color in the area under the line (optional).

Prepare a set of index cards in different colors for each team member. Write the name of one measure on all the cards of a single color. So if you are measuring team values (as in Figure 5.5, on the page before), all the green cards would have "Communication" written on one side, all the blue cards would have "Courage", and so on. Each team member receives a set of cards that includes each factor measured.

Examples

Team Radar is a subjective measure that's intended to generate discussion. This is especially useful when you suspect there's no common definition or criteria to measure against.

For example, one team used a radar to examine how team members perceived their use of a number of engineering practices, including refactoring. One team member rated their refactoring 8; another rated it at 3. During the discussion that followed, it became clear that each had different ideas on when to refactor. Further, the team member who rated her refactoring low was upset with the team member who rated his refactoring high because he was "slacking off by not refactoring enough." By the end of the retrospective, the team arrived at a common definition. Over the next few iterations, the team was more consistent in when they refactored, and resentment faded.

6.2 Activity: Force Field Analysis

Use this in conjunction with an activity that suggests possible changes while generating insights for a release or project retrospective. Use this as part of a planning exercise while deciding what to do.

Purpose

To examine what factors in the organization will support a proposed change and which will inhibit the change.

Time Needed

Forty-five to sixty minutes depending on the complexity of the issue and the size of the group.

Description

The team defines a desired state they want to achieve. Small groups work to identify the factors that could either restrain or drive the change they want. The factors are listed on a poster; then the group assesses the strength of each supporting factor relative to the other supporting factors and repeats the process for inhibiting factors. The team discusses which factors they can influence—either by increasing the strength of a supporting factor or by reducing the strength of an inhibiting factor.

Steps

1. Introduce the activity by saying "If we want this change to succeed, we need to understand more about the factors that will support or inhibit the change."
2. Describe the process.

   Break into small groups (no more than four).

   "Each group works for __ minutes to identify factors that will drive or support the change."

   "We'll do a round-robin report of what you discover and post the results. Then we'll repeat the process for restraining or inhibiting factors."

"After we've listed both sets of factors, we'll assess their relative strength and discuss what course of action would be most helpful for implementing the change we want."

3. Monitor time and the activity level.

   While the groups are working, prepare a flip chart like the one pictured in Figure 6.2, on the next page (but don't fill in the factors yet).

4. When the group is finished with the first task (identifying supporting or driving factors) collect the information the small groups have generated in round-robin fashion, asking for one factor at a time. There's no need to repeat duplicates; collect only the unique factors.

5. Repeat for restraining or inhibiting factors.

6. Bring the whole group back together, and examine each factor and gauge its strength relative to the other factors. Draw a line toward the center arrow indicating relative strength. Do this first for driving and then for restraining factors.

7. Examine the factors for most effective actions:

   - Ask the group how they can strengthen driving factors or mitigate restraining factors.
   - Ask the group whether enhancing driving factors or reducing restraining factors is more likely to achieve the desired state.

Materials and Preparation

Flip chart paper or a white board. Markers.

Identify an issue to analyze, perhaps from a list of proposed improvements or another generating insights activity, such as Five Whys or Fishbone.

Example

Force Field Analysis is another tool to ensure that the changes your team identifies in their retrospective actually happen. Combine creating the Force Field Analysis graph with a discussion of influence and control. What can the team directly control to make a change? What can't they control, and where are their points of influence in the situation? Most teams have more ability to influence conditions than they

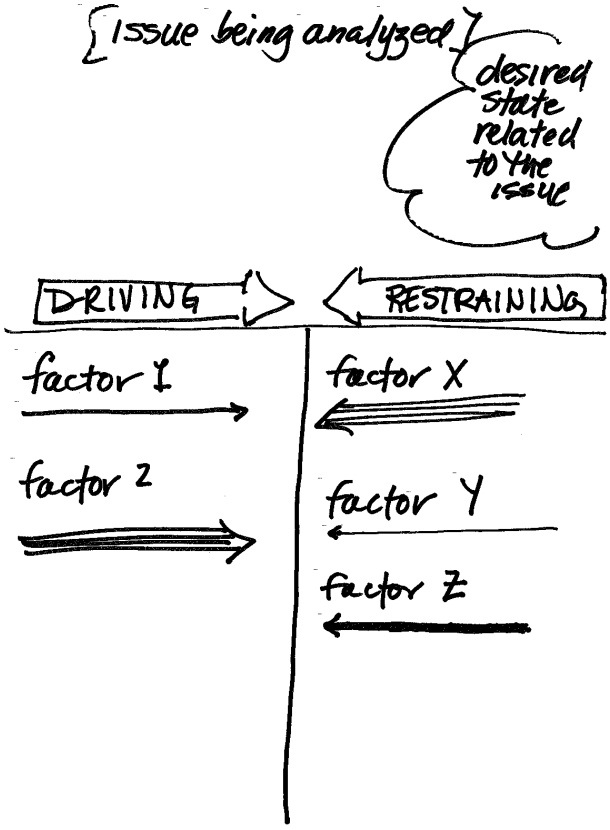


Figure 6.2: Force Field analysis helps the team look at factors affecting a proposed change.

realize; however, a team needs to think about the most effective ways and times to use their influence. Force Field Analysis can help them discern points of greatest leverage and, sometimes, help them see that changing a situation may require more effort than the outcome they desire will be worth. Other times, they may see the forces allied against them and decide to tackle the issue anyway.

One team came into their retrospective wanting to change the way they interacted with the product owner. They were dissatisfied with the limited contact and communication that occurred during the iteration. The product owner answered questions, but only after several days had passed.

Before they analyzed the situation by drawing a Force Field Analysis poster, they understood that the product owner's travel schedule and times of availability were outside their control. Afterward, they also saw they could exert influence best by explaining their concerns to the VP of Marketing, another person with a crazy travel schedule.

They decided that tracking down the VP would take more team effort than they could afford. Instead, they made plans to get the most out of the few product owner contacts available to them.

6.4 Activity: Fishbone

Use this to generate insights in a longer iteration, release, or project retrospective.

Purpose

Look past symptoms to identify root causes related to an issue. Look for reasons behind problems and breakdowns.

Time Needed

Thirty to sixty minutes.

Description

The team identifies factors that are causing or affecting a problem situation and then looks for the most likely causes. After they've identified the most likely causes, they look for ways they can make changes or influence those factors.

Steps

1. Draw a fishbone diagram (see Figure 6.3, on the following page) and write the problem or issue at the fish's head. Include the five W's—What, Who, When, Where, and Why. Label the "bones" of the fish with categories.

   Typical categories are as follows:

   - Methods, Machines, Materials, Staffing (formerly known as Manpower)
   - Place, Procedure, People, Policies
   - Surroundings, Suppliers, Systems, Skills

   You can use these in any combination, or the team can identify their own categories.

2. Brainstorm factors within each category. Prompt by asking "What are the [fill in a category name here] issues causing or affecting [fill in the problem here.]" Repeat this for each category. Write the issues along the bones, or have people write them on small sticky notes and stick them to the fishbone diagram.

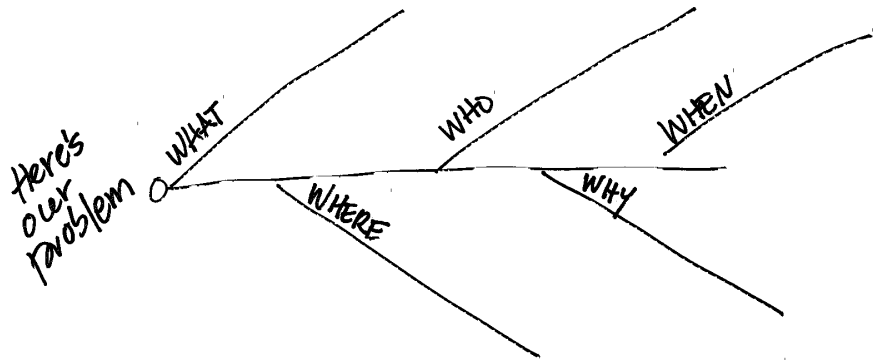


Figure 6.3: Fishbone is a way to look at root causes.

3. Continue asking "Why is this happening?"

   Add more branches off the bones as needed.

   Stop when the causes are outside the team's control or influence.

4. Look for items that appear in more than one category. These may be the most likely causes. Engage the group in looking for areas where they can make a difference.

   Use the results in the next phase, Decide What to Do.

Materials and Preparation

Markers, sticky notes.

Define the problem statement. Include the five W's— What, Who, When, Where, and Why—to the extent they are known. Draw the fishbone diagram on a flip or white board. Make a list of the sample categories.

Examples

Use a Fishbone activity to dig into the root causes of a problem, but don't stop there. A fully branched and labeled diagram is not a deliverable of the retrospective.

If you suspect that a lot of what will come up in the retrospective may be due to issues outside the team's control, digging into all the problem sources may drain the team's energy. Choose a different method.

When the issues are more local to the team and under their direct control, the team may be energized by tackling the fishbones.

For example, during a two-week iteration, the build broke five times. The retrospective leader knew the team was frustrated by it, and the broken build would be a prominent topic in the retrospective. He introduced the Fishbone activity with bones labeled “Skills”, “Systems”, “Surroundings”, and “Staffing”.

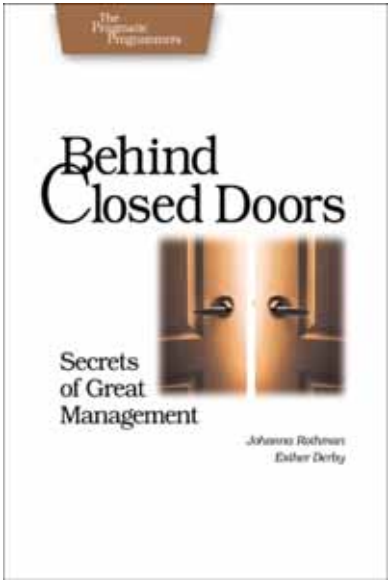
Two or three team members worked in small groups to focus on writing sticky notes for each bone. They covered the “fish” with scales of notes.

When they stepped back to read the notes, they saw two root causes—inexperienced team members working alone (showed up in both Skills and Staffing) and writing new code while waiting for the build to compile (showed up in both Systems and Surroundings). Everyone immediately agreed on a commitment to mentor and pair with newer team members. They identified the second cause as needing more attention and decided to include it as a topic of action planning.

Also by Esther Derby

If you liked this book by Esther Derby and Diana Larsen, you might also enjoy the following title by Johanna Rothman and Esther Derby.

Behind Closed Doors



Secrets of Great Management.

You can learn to be a better manager—even a great manager—with this guide. You'll follow along as Sam, a manager just brought on board, learns the ropes and deals with his new team over the course of his first seven weeks on the job.

From organizing who works on what project when, to helping team members grow and prosper, you'll be there as Sam makes it happen for the entire team.

You'll find powerful tips covering:

- Delegating effectively
- Using feedback and goal-setting
- Developing influence
- Handling one-on-one meetings
- Coaching and mentoring
- Deciding what work to do-and what not to do
- . . . and more.

Full of tips and practical advice on the most important aspects of management, this is one of those books that can make a lasting and immediate impact on your career.

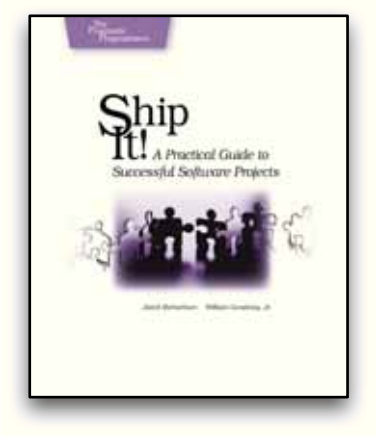
Behind Closed Doors Secrets of Great Management
Johanna Rothman and Esther Derby
(192 pages) ISBN: 0-9766940-2-6. $24.95

Competitive Edge

We've got you covered—whether it's titles to maintain your competitive edge, or the latest cutting-edge technologies.

For a full list of all of our current titles, as well as announcements of new titles, please visit www.pragmaticprogrammer.com.

Ship It!



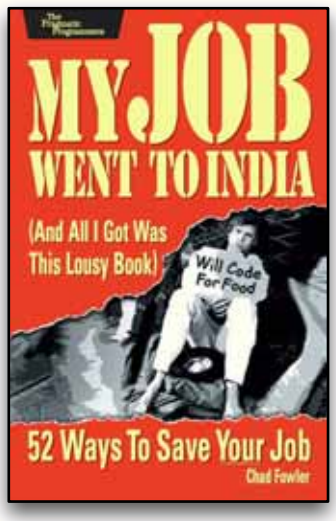
Agility for teams. *Ship It!* lets you do just that: on time and on budget, without excuses. You'll see how to implement the common technical infrastructure that every project needs along with well-accepted, easy-to-adopt, best-of-breed practices that really work, as well as common problems and how to solve them.

Ship It!: A Practical Guide to Successful Software Projects
Jared Richardson and Will Gwaltney
(200 pages) ISBN: 0-9745140-4-7. $29.95

My Job Went to India

World class career advice. The job market is shifting. Your current job may be outsourced, perhaps to India or eastern Europe. But you can save your job and improve your career by following these practical and timely tips. See how to: • treat your career as a business • build your own brand as a software developer • develop a structured plan for keeping your skills up to date • market yourself to your company and rest of the industry • keep your job!

My Job Went to India: 52 Ways to Save Your Job
Chad Fowler
(208 pages) ISBN: 0-9766940-1-8. $19.95



Visit our secure online store: http://pragmaticprogrammer.com/catalog

Cutting Edge

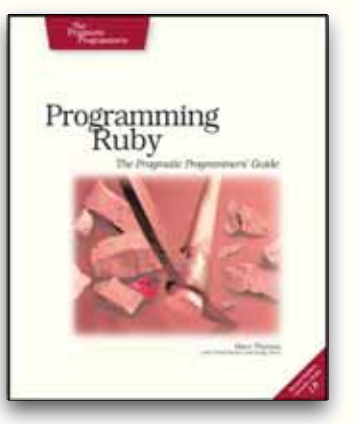
Learn how to use the popular Ruby programming language from the Pragmatic Programmers: your definitive source for reference and tutorials on the Ruby language and exciting new application development tools based on Ruby.

The *Facets of Ruby* series includes the definitive guide to Ruby, widely known as the PickAxe book, and *Agile Web Development with Rails*, the first and best guide to the cutting-edge Ruby on Rails application framework.

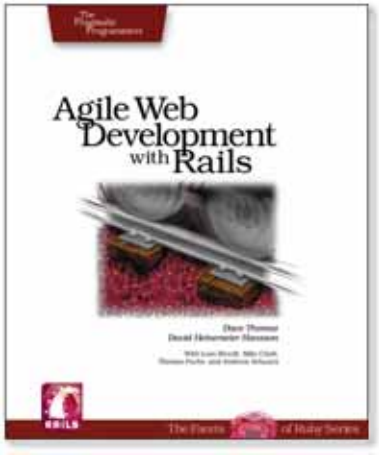
Programming Ruby (The PickAxe)

The definitive guide to Ruby programming. • Up-to-date and expanded for Ruby version 1.8. • Complete documentation of all the built-in classes, modules, methods, and standard libraries. • Learn more about Ruby's web tools, unit testing, and programming philosophy.

Programming Ruby: The Pragmatic Programmer's Guide, 2nd Edition
Dave Thomas with Chad Fowler and Andy Hunt
(864 pages) ISBN: 0-9745140-5-5. $44.95



Agile Web Development with Rails



A new approach to rapid web development. Develop sophisticated web applications quickly and easily • Learn the framework of choice for Web 2.0 developers • Use incremental and iterative development to create the web apps that users want • Get to go home on time.

Agile Web Development with Rails: A Pragmatic Guide
Dave Thomas and David Heinemeier Hansson
(570 pages) ISBN: 0-9766940-0-X. $34.95

Visit our secure online store: http://pragmaticprogrammer.com/catalog

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style, and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help programmers stay on top of their game.

Visit Us Online

Agile Retrospectives's Home Page
`pragmaticprogrammer.com/titles/dlret`
Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates
`pragmaticprogrammer.com/updates`
Be notified when updates and new books become available.

Join the Community
`pragmaticprogrammer.com/community`
Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy
`pragmaticprogrammer.com/news`
Check out the latest pragmatic developments in the news.

Buy the Book

If you liked this PDF, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: `pragmaticprogrammer.com/titles/dlret`.

Contact Us

Phone Orders:	1-800-699-PROG (+1 919 847 3884)
Online Orders:	`www.pragmaticprogrammer.com/catalog`
Customer Service:	`support@pragmaticprogrammer.com`
Non-English Versions:	`translations@pragmaticprogrammer.com`
Pragmatic Teaching:	`academic@pragmaticprogrammer.com`
Author Proposals:	`proposals@pragmaticprogrammer.com`