



Chapter 4

Iterating Toward Agility

Historically, when an organization needed to change, it undertook a “change program.” The change was designed, had an identifiable beginning and ending, and was imposed from above. This worked well in an era when change was necessary only once every few years. Christopher Avery has written, “I think in the 1960s and 1970s this approach was probably more frequently successful than it has been in the 1990s and today because the frequency of change has intensified as competition has become global, and the model has broken down” (2005, 18). Avery continues by saying that “if the changes are coming so fast and furious that programmed change won’t work, perhaps we have to arrange ourselves (organizationally speaking) to digest many more smaller changes on a continual basis” (20).

Whether you are just starting to adopt Scrum or you are at the point where you are ready to fine-tune your use of Scrum, you should manage the effort in an agile way. Following an iterative transition process—making small changes on a continual basis—is a logical way to adopt a development process that is itself iterative. Doing so will be much more likely to result in a successful and sustainable transition. This is why I believe that the effort of adopting Scrum is best managed using Scrum itself. With its iterative nature, fixed timeboxes, and emphasis on teamwork and action, it seems best suited to manage the enormous project of becoming and then growing agile with Scrum.

In 2004, the leaders of Shamrock Foods realized that change was coming too quickly in their industry. As one of the ten largest food distributors in the United States, Shamrock had for 20 years used a conventional, top-down strategic planning process, dedicating months each year to creating a 5-year plan that was out of date before the ink dried. To address this problem, CEO Kent McClelland abandoned the company’s 20-year-old approach and began to apply a Scrum-based iterative strategic planning process.

Shamrock’s process revolved around quarterly strategic “scrums” [sprints]: Team members met at an offsite location for a day to evaluate the company’s performance against the action plans from the previous quarter. We asked them to identify the most important things they had learned about the company’s strategy





62 | Chapter 4 Iterating Toward Agility

since the previous meeting and to suggest how those insights should be integrated in the strategy going forward. The group created new action plans for the upcoming period. In addition to the quarterly scrums [sprints], the participants met every year for three days, during which time people were asked to step further back and revisit the company's strategic assumptions. (McFarland 2008, 71)

Forty-five managers and employees participated in these sprints and were chosen to represent each division and functional area. At the start of each quarterly sprint, this group selected up to a handful of key areas in which they agreed the company should improve. These were referred to as themes. Because Shamrock was applying Scrum to an organizational improvement effort rather than software development, the themes represented broad business goals. Examples included increasing revenue on Shamrock's house brands, improving how it serviced large customers like Burger King, and improving the company's ability to recruit, retain, and develop good talent.

Many corporate improvement initiatives fail because plans are not made specific and actionable. Because they were using Scrum, Shamrock employees went beyond just identifying themes for improvement: "Planning participants created and prioritized a handful of specific and measurable strategic initiatives that would advance each strategic theme. Then they built detailed action plans and set measurable outcomes they thought could be achieved within 90 days" (McFarland 2008, 71).

Not only does the Shamrock story illustrate the broad applicability of Scrum, it serves as an example of how Scrum can be used to manage an organizational improvement effort. In this chapter, we look at how to use Scrum first to adopt Scrum and then to continuously improve by engaging communities of like-minded employees, such as the 45 people who guided Shamrock's improvement effort.

The Improvement Backlog

Just as Scrum development projects use product backlogs, you should use an *improvement backlog* to track the effort of adopting Scrum in your organization. An improvement backlog lists everything that the organization could do better in its use of Scrum. When IBM began to adopt Scrum, its improvement backlog included the following items:

- Increase the number of teams using Scrum.
- Increase adoption of test automation.
- Enable teams to implement continuous integration.





- Figure out how to make sure each team has a product owner.
- Determine how we're going to measure the impact of adopting Scrum.
- Increase the use of unit testing and test-driven development.

Improvement backlogs, such as the one shown in Table 4.1, are dynamic, with items coming and going as they are thought of, completed, found unnecessary, and so on. Much of what we discussed in Chapter 2, "ADAPting to Scrum," will find its way onto an improvement backlog. If you're just starting with Scrum, your improvement backlog will emphasize creating awareness and desire. If the transition is already well underway, your improvement backlog may contain more items around developing the ability to do Scrum well, to promote successes, or to transfer it to other groups. Similarly, decisions about which patterns to use, as described in Chapter 3, "Patterns for Adopting Scrum," can create items on an improvement backlog.

A small department or single-project transition may involve a single improvement backlog. But when Scrum is being adopted across a large site, department, or organization, the transition effort becomes large enough that multiple improvement backlogs are used, each of which is created by a community of individuals who are passionate about improving the organization in a particular way. There may be, for example, a community and associated improvement backlog for figuring out how best to do automated testing on Scrum projects, another for developing and becoming great ScrumMasters, and so on.

Additionally, in a large transition effort, there is what might be considered a master improvement backlog, which is maintained by the group that guides the organization's overall transition. It is to that group that we turn our attention next.

The Enterprise Transition Community

The small group that initiates, encourages, and supports an organization's effort to introduce and improve at Scrum is known as the Enterprise Transition Community, or ETC.¹ The Enterprise Transition Community exists to create a culture and environment where change can be released by those who are passionate about the success of the organization and where success leads to more passion from more people. The ETC does this not by imposing changes on the organization but by guiding groups who are implementing changes, by removing obstacles to doing Scrum well, and by creating energy and excitement for the change.

The members of the ETC, who usually number no more than a dozen, come from the highest level involved in the transition to Scrum. If a company is adopting Scrum organization-wide, the ETC should include senior people from

¹ The acronym ETC is consistent with Ken Schwaber's in *The Enterprise in Scrum*, although he refers to it as the "Enterprise Transition team" (2007).



**64** | **Chapter 4** Iterating Toward Agility

engineering or development plus vice presidents of groups such as product management, marketing, sales, operations, human resources, and so on. For a departmental adoption of Scrum, the ETC may include the vice president of engineering along with the heads of QA, development, architecture, interaction design, database, and so on. The key here is that the ETC is made up of the most senior people for the level at which the transition is occurring.

TABLE 4.1

An improvement backlog is a list of capabilities to be developed, work to be performed, or issues to be addressed within the organization.

Item	Responsible	Note
Create a Scrum Office (like a Project Management Office) where teams can get help.		Jim (CTO) to talk this up at monthly development meeting. Let's see if there's any interest.
Establish an internal program for developing ScrumMasters.		How do we identify good internal candidates? How do we develop them?
Collect and disseminate Scrum success stories in our company.	SC	Savannah has expressed interest in this.
Develop a continuing education program internally.		Consider quarterly open space meetings. Identify and contact industry experts for one-hour lunch meetings.
Start doing lots of automated unit testing (even if it's not test-first) and using FitNesse.		The Scrum team that makes the most progress on this (as voted on by everyone in the department) can have all members attend next summer's Agile conference.
Help a community team form to decide how much up-front architecture is enough.	TG	Tod to start soliciting volunteers but says he can't commit to any goals for it until next quarter.
Resolve dispute with facilities over rearranging second floor cubicles.	JS	Jim to talk to Ursula in facilities about budget for this.
Craft message on why we're adopting Scrum; have Jim discuss it at his monthly meeting.	JS	Next meeting is 25 March.

Sometimes Scrum is introduced into an organization in a grassroots manner. One team tries Scrum and successfully completes a project, others become interested, and Scrum spreads from there. In this situation, an ETC is usually formed





spontaneously by some of these early Scrum advocates who ask their boss to be allowed time to help other teams learn Scrum. At some point, impediments arise that need the help of that boss, who then joins the ETC. Alternatively, in an enterprise-wide Scrum adoption, the ETC is usually formed more deliberately when the decision is made to widely adopt Scrum.

As an example of an ETC, consider the case of Farm Credit Services of America, a lending and financial services cooperative that works with farmers in the American Midwest. As part of adopting Scrum, Farm Credit formed an Enterprise Transition Community it calls the Agile Champions Team (ACT). The 16 or so individuals on the ACT participate on the team for between 6 to 24 months depending on their role in the organization and ability to commit time to the team. Because the transition at Farm Credit covers the organization's entire information services and business departments, ACT members are chosen to equally represent all functions involved. The Farm Credit ACT meets every other week for two hours and augments those meetings with occasional longer offsite meetings.

Comprising both formal and informal leaders, the ACT often works on issues that arise between the information services department and the broader business. It has resolved issues related to a lack of stakeholder involvement in projects, the proper use and meaning of deadlines, and executive leadership misperceptions of what agile is and can do for the company. Quinn Jones is a software developer at Farm Credit who served what he calls a six-month "tour of duty" on the ACT. He says, "One of the best things to come out of the Agile Champions Team is the wide-open, smack-down brown bag sessions where all are welcome to ask questions and share knowledge. These meetings have also helped uncover root challenges in agile, which could then be addressed by the ACT."

- ❑ Write a preliminary improvement backlog by convening a 30- or 60-minute meeting. Invite either your team members, a few people you know will be interested, or the whole department. Brainstorm things that you'd like to see improved. Conclude the meeting by asking if there is sufficient passion to pursue just one or two of the items, and then start with those.

THINGS TO
TRY NOW

ETC Sprints

Because the ETC uses Scrum, it makes progress in sprints, exactly like a Scrum development team would. Each ETC sprint begins with a planning meeting and ends with a review and retrospective. These meetings are completely analogous to those held by Scrum development teams and often have the same problems. Thomas Seffernick, of KeyCorp, a large U.S. financial institution, participated in the first sprint review of his organization's ETC, which it called an Agile Enablement Team. He recalls how that team made a mistake common to many new





66 | Chapter 4 Iterating Toward Agility

Scrum development teams—talking about its plans rather than demonstrating its progress.

That first Agile Enablement [ETC] sprint review was painful as leaders stood up and described their plans to remove the impediments they volunteered to address. The message was clear—plans are good, but results count. The dynamic of those reviews changed from that point, and results became the focus. (2007, 202)

Some ETCs hold daily scrums, and I think that is a good practice. But, I am not as insistent upon this as I am with a Scrum development team. The work being done by members of an ETC is not as tightly interwoven as the work of a development team, making a daily scrum a great thing to do but not essential. Similarly, ETC members are rarely full-time. Most have demanding jobs already, and in many cases it is beneficial for them to remain in their jobs. A development director who stays in that position, for example, can likely remove more organizational impediments than a development director who steps out of that position to serve on the ETC.

The length of an ETC sprint is up to its members. However, in my experience two-week sprints work best. This is also the sprint length recommended by Ken Schwaber (2007, 10). Elizabeth Woodward, a member of the ETC that is guiding the large-scale adoption of agile at IBM, describes the company's sprint length experience.

We've used both two-week and four-week sprints. And, so far, the greatest success we've seen is with those on two-week sprints. I believe the reason is that the "deliverables" demonstrate momentum and visible progress. We capture the efforts from each community in a brief digest—a nice e-mail message that people can read in about fifteen minutes.

The Sponsor and Product Owner

Most successful Scrum adoptions have been initiated or driven by an identifiable sponsor, who is a senior person in the organization responsible for the success of the transition. Salesforce.com's highly successful large-scale transition was sponsored by company cofounder Parker Harris. As the executive vice president of technology, Harris was well positioned to champion a change that would dramatically alter how everyone in the Salesforce.com development organization worked.

The transition's sponsor should come from the same level in the organization at which the transition is being planned. Salesforce.com needed an executive as





its sponsor because it was doing an enterprise-wide transition. If you are involved in a departmental transition, a department-level leader is an appropriate choice.

The sponsor is also the product owner for the ETC. This means that sometimes an ETC will have a product owner with little direct experience with Scrum. That's OK. Like all product owners, the sponsor of the ETC can fulfill the role by calling on other ETC members for help. As the ETC's most senior member, the sponsor will play a significant role in communicating about the transition effort, but this person does not need to be the sole source of the vision.

Primavera learned the importance of a strong sponsor when it adopted Scrum. Bob Schatz and Ibrahim Abdelshafi, technology executives within Primavera at the time, write about the importance of a sponsor's support.

Adopting agile, or implementing any significant change, requires an executive's sincere support. It can be a bumpy ride until things settle down, and having executive support lets the learning take hold despite any problems or failures. (2005, 38)

It is critical that the sponsor demonstrate commitment to the transition effort by participating on the ETC. Good sponsors do not initiate a transition, proclaim support for Scrum, and then remove themselves from the effort of getting there. If a sponsor is not committed, others will not be either. Scrum coach and author of *Collaboration Explained*, Jean Tabaka considers a checkbook-only commitment from a sponsor to be one of the most likely reasons a Scrum adoption might fail: "Agile adoption requires a passionately engaged sponsor willing to make tough organizational changes that serve agile teams and their success" (2007).

Although it would be fair to characterize ETC members as leaders of the Scrum adoption effort, theirs is not what we think of as conventional leadership. Writing in *Harvard Business Review*, internationally respected management author Henry Mintzberg describes the necessary type of leader.

Communityship requires a more modest form of leadership that might be called *engaged* and *distributed* management. A community leader is personally engaged in order to engage others, so that anyone and everyone can exercise initiative. (2009, 141; emphasis his)

Mintzberg goes on to say that during an organizational change like adopting Scrum, "we need *just enough leadership*—leadership that intervenes when appropriate while encouraging people in the organization to get on with things."





OBJECTION

"The sponsor of our transition project says he's committed, but he's unable to come to any meetings or to put any time into the effort. He gives us anything else we need, but we can't get any of his time."

You probably have the wrong sponsor. Although his willingness to support the transition in other ways is admirable, a successful Scrum transition requires some of the sponsor's time. You don't want to lose this powerful ally, but you may need to look for a different sponsor. Alternatively, you may want to negotiate with your sponsor for a small amount of his time. The ETC can then prioritize how that time should be spent. It could perhaps be in meetings or as a public supporter of the transition in other forums.

Responsibilities of the ETC

An ETC is a working group. It is not a steering committee. During sprint planning, the ETC commits to completing some amount of work and demonstrating it at the end of the sprint. However, even more important than the tangible things the ETC accomplishes is that it ignites the interest of others. Members of the ETC can only achieve so much themselves. They will need to rely on others in the company to do most of the work of adopting Scrum and becoming agile. Change management experts Edwin Olson and Glenda Eoyang concur.

In a self-organizing system, the leader has an important role to play, but creative and long-lasting change depends on the work of many individuals at many different levels and places in the organization. (2001, 5)

One of the most important jobs of the ETC is creating energy around the adoption of Scrum. Not everyone will be excited by the change, of course. But the ETC needs to ignite the passion of those who will work to make adopting Scrum successful. ETC members do this by showing their own enthusiasm and by participating in constructive dialogue about the changes occurring. To ignite the passion of others in the organization so that they become involved in the type of creative and long-lasting change needed to adopt Scrum, the ETC is responsible for the following:

- **Articulate the context.** Beyond conveying a vision of the organization's agile future, the ETC must also help employees both understand the need to change and develop a desire to change. They do this by articulating the context of the change: Why? Why now? Why Scrum? Members of the ETC use their seniority, personal credibility, and more to get others to understand the answers to these questions.





- **Stimulate conversation.** All sorts of good things happen when people talk. Debating the merits of various technical practices, sharing success stories, probing reasons for failure, and other discussions will generate ideas.
- **Provide resources.** Adopting Scrum will take time, effort, and money. For example, individuals who are trying to figure out how to be more agile (say, learning how to write automated unit tests on a complicated code base) may need to be granted time away from their development projects. Because the ETC includes the most senior people involved in the transition, the ETC is in a position to ensure that both time and money are available.
- **Set appropriate aspirations.** Change efforts with clearly defined and truly transformational goals are ten times more likely to succeed (McKinsey & Company 2008). The ETC is responsible for setting and communicating appropriate goals for the transition, which may (and probably should) change over time as the organization improves. The ETC may establish goals such as moving from one annual release to quarterly releases, a 50% decrease in post-release defect rate, or so on.
- **Engage everyone.** Scrum has long tentacles and will reach into many areas of the organization. The ETC makes sure that the transition effort does not become narrowly focused on just one group. Within the groups that are affected, broad participation is encouraged.

SEE ALSO

Advice on appropriate metrics for measuring your progress is offered in Chapter 21, "Seeing How Far You've Come."

Additional Responsibilities

Beyond encouraging people to engage in the transition, the ETC has the following additional responsibilities:

- **Anticipate and address people issues.** The ETC should try to anticipate which groups or individuals are going to struggle the most with the changes brought about by Scrum and proactively work with them. The cross-functional makeup of the ETC helps in this regard as it allows the group to see problems from multiple perspectives.
- **Anticipate and remove impediments.** Members of the ETC are responsible for removing any organizational impediments to adopting Scrum or doing it well. Beyond merely removing impediments it is informed of, the ETC should try to anticipate obstacles and remove them before they cause problems.
- **Encourage a simultaneous focus on practices and principles.** Adopting Scrum involves incorporating new practices and valuing new principles. An organization cannot adopt the practices without the underlying principles, nor can it adopt the principles without the practices. An effective ETC looks for imbalances in how each is being adopted. If one is being





70 | Chapter 4 Iterating Toward Agility

accepted faster than the other, the ETC can bring them back in line by directing conversation, attention, and resources toward the laggard.

If an ETC performs these tasks well, not only will it be moving the organization forward on its own, but it also will have generated interest and excitement among others in the organization. To harness that passion, individuals with a common interest in improving the organization in a particular way (perhaps its adoption of automated testing) come together, form a community of their own focused on that improvement, and then run their own sprints. These communities are known as *improvement communities* and are the topic of the next section.

OBJECTION

"I can't get the organizational backing to create an ETC. Can I still transition to Scrum?"

Yes. Start with whatever sphere of influence you do have. Get your team to do Scrum. If it is successful, people will notice. Perhaps another team will want to do Scrum and ask for advice. Or a manager will get interested. As people get interested, start the community informally as just a few people who get together occasionally to talk about how Scrum is going and what could be done better. A grassroots approach is very feasible but will take longer to spread.

THINGS TO TRY NOW

- ❑ If you don't already have an ETC or equivalent group, identify several people who ought to be on the ETC. If you are one of them, begin forming this group. If you are not, share the idea of an ETC and improvement communities with others in your organization who can help form these groups.

Improvement Communities

An improvement community (IC) is a group of individuals who join together to work collaboratively to improve the organization's use of Scrum. An IC may form when individuals notice an item on the ETC's improvement backlog and decide to work together to achieve that goal. Or an IC may form because individuals see and are passionate about an improvement opportunity that hasn't made the ETC's radar yet. IBM, for example, has five ICs, which are focused on test automation, continuous integration, test-driven development, the role of the product owner, and the general use of Scrum itself.



The Enterprise Transition Community and improvement communities I am referring to are specialized types of what are known as communities of practice (Wenger, McDermott, and Synder 2002). A community of practice is a group of like-minded or like-skilled individuals who voluntarily come together because of their passion and commitment around a technology, approach, or vision. We will see other types of communities of practice throughout this book. They will be thoroughly discussed in Chapter 17, "Scaling Scrum."

NOTE

Graphically, the relationship between an organization's one ETC and its multiple ICs can be seen in Figure 4.1. The ETC guides the transition process; it does not direct or manage it. A big part of its role is fostering an environment in which ICs form and dissolve organically in pursuit of improving how the organization builds products.

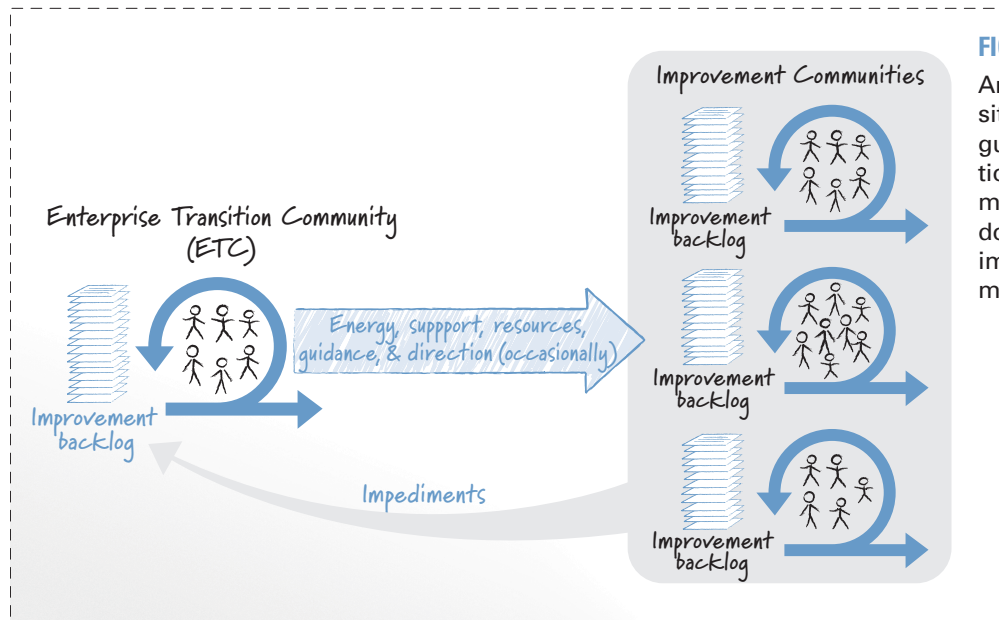


FIGURE 4.1

An Enterprise Transition Community guides the adoption of Scrum, but most of the work is done by multiple improvement communities.

This approach should be scaled up or down depending on the size of the organization undertaking the transition. A software development department of 30 people may have an ETC of 5 people and nothing more. A company-wide transition for a department of 200 developers may have a 10-person ETC (including representatives from groups outside development) plus a handful of improvement communities at any time. Things can scale from there as needed; IBM, for example, has over 800 people in some of its improvement communities.

Most participants in an IC spend only a small part of their time engaged with the community. They may read postings to its discussion list, add a comment on



72 | Chapter 4 Iterating Toward Agility

a wiki, and nothing more. The amount of time an IC member spends on the community is determined by each individual, the person's boss, or organizational culture.

OBJECTION

"Scrum teams are supposed to be self-organizing. Doesn't an ETC conflict with this? Shouldn't teams get to decide what they want to improve at?"

Self-organization occurs in response to a challenge taken on by a group of individuals. For a development project, the company may tell a team, "Develop this software to run faster and take less memory than the current version and do it two months faster than we've done in the past." Individuals then organize themselves around how to achieve that goal. It is no different with the ETC. An ETC states what it would like to see improved but not necessarily how to achieve that improvement. The how is left up to the improvement communities or Scrum teams.

Additionally, keep in mind that an ETC's biggest goal is to create an environment such that improvement communities identify their own goals and form spontaneously to address them. We will look at self-organization in detail in Chapter 12, "Leading a Self-Organizing Team."

Catalysts for Improvement

Communities, when used as part of the effort to adopt and get good at Scrum, become catalysts for improvement. Consider the case of Google, where improvement communities are called "grouplets." Google's Testing Grouplet was formed "to drive adoption of developer testing" (Striebeck 2007). Bharat Mediratta founded the community and describes its activities.

We started with engineers from all over the company meeting every couple of weeks to brainstorm. Slowly, over time, we started turning into activists, planning to actually start improving things. We started building better tools and giving informal talks to different technical groups. (Mediratta 2007)

Notice that although this community met initially to brainstorm, they soon found themselves as activists with plans for actual improvements. Improvement communities act. This is why they aren't called task forces, work groups, committees, or any of the other terms that too often bring to mind ineffective groups. If the Google Testing Grouplet had merely created presentations on the benefits of developer testing, or if it had chosen to convince a powerful vice president to mandate developer testing, its efforts would have been fruitless.





What the testing community at Google did instead was find direct and immediate ways to help teams. Mediratta recalls how, in addition to building tools, the community found a unique way of providing concrete, short examples and advice about testing.

One day, toward the end of a long brainstorming meeting, we came up with the idea of putting up little one-page stories, called episodes, in bathroom stalls discussing new and interesting testing techniques. Somebody immediately called it “Testing on the Toilet,” and the idea stuck. (2007)

The most effective communities are usually those that form not in response to management dictate but because company culture or the ETC has created an environment in which communities can naturally emerge. J. F. Unson, a coach at Yahoo! during its large-scale Scrum rollout, says this is exactly what happened at one of Yahoo!’s remote facilities.

At Yahoo!, in our Santa Monica campus, all the entertainment agilistas started a monthly ScrumMaster lunch. This happened organically as Scrum started to grow in the organization, without having the agile group [ETC] pushing it. (2008)

Not all communities will form in such an organic manner, of course. Especially during the early weeks or months of adopting Scrum, the ETC will need to encourage an improvement community to form by highlighting the importance of a goal and then hoping a community forms around that goal. Occasionally, an ETC may need to go so far as to ask someone to form a community around a specific goal.

Two Metrics for Effectiveness

Professor Jeffrey Goldstein has written, “Change does not need to be imposed; it simply needs to be released” (1994, 32). You can gauge how well the ETC is doing at releasing change in two ways:

1. The number of improvement communities that have formed without a direct request from the ETC
2. The percentage of such improvement communities to the total number of improvement communities

If the number of spontaneously formed improvement communities is high, and especially if these represent a majority of the total number of communities, this indicates strong interest in Scrum and the changes it is creating. If these metrics are increasing or remain high over time, the organization is well on its way to becoming agile. You should, of course, look at other metrics. These are just two that I like.



**74** | **Chapter 4** Iterating Toward Agility**An Improvement Community Sprint**

As you might suspect, ICs perform their work in sprints as well. As with the ETC, each IC can select its own sprint length, but two weeks is the recommended length. An IC that was formed spontaneously will usually serve as its own product owner, with members of the community electing to devote their time to the improvements they are the most passionate about. An IC that was formed in response to an ETC-identified goal, on the other hand, will usually work with a member of the ETC as its product owner to plan a sprint.

That being said, an improvement community does not exist to serve the ETC. It exists to serve its customers: the Scrum development teams who are building products or systems. Although an ETC member will act as product owner for some improvement communities and will serve as the official product owner for the sprint reviews, you should expect members of interested development teams to be active participants as well. Additionally, the wise ETC understands that the best results will be achieved when improvement communities are given broad latitude in achieving their goals. In practice, this means an IC, even one formed in response to ETC-identified goals, will be responsible for prioritizing its own work, while balancing the needs of the organization to improve in particular ways and its members' passion for working on those issues.

During its sprint planning meeting, each improvement community selects one or more things it can commit to completing during the sprint. If an improvement community has formed in response to a specific goal of the ETC, sprint planning begins by taking an item from the ETC's backlog and breaking it down into smaller items that will be placed on the improvement community's improvement backlog. The best way to see this is with an example.

The ETC improvement backlog shown in Table 4.1 on page 64 includes the item, "Establish an internal program for developing ScrumMasters." An improvement community formed a month after the ETC put that on the improvement backlog and made it known to the rest of the company that creating such a program would be valuable. There were three people in the community initially, but that was plenty to make progress toward this goal. In their first sprint planning meeting, they discussed the ETC's goal ("Establish an internal program for developing ScrumMasters") and created their own improvement backlog of what they would do to achieve this goal, which is shown in Table 4.2.

Also during sprint planning, the community members took some of the items in Table 4.2 and identified the tasks necessary to complete each. For example, for the final item in Table 4.2 (working with local groups to share the expense of bringing in speakers), the community identified the following tasks:

- Search web to see what user groups are in our area.
- Create budget of expenses.





- Send e-mail to internal distribution lists to see if anyone here is connected to these groups.
- Set up phone calls to introduce ourselves and what we're doing.
- Conduct phone calls. See if any groups have previously split the cost of bringing a speaker into town with another company. See if any will work with us on this.
- Meet with Susan to go over budget and get approval.

What	Note
Figure out how to identify good candidates to become ScrumMasters (in addition to those who ask to participate in this program).	
Establish an internal mentoring program.	
Develop some internal classroom training. Which courses? Who can teach them? Develop our material, or can we license it?	
Determine which classes we can teach internally.	
Get budget for next year for external coaching. How many days? At what expected daily rate?	James has already asked for rates from three coaches.
See what we can do with local user groups to share the expense of bringing in speakers.	Savannah has contact in local Scrum lunch meetup group.

TABLE 4.2

An improvement community's backlog for establishing an internal program to develop ScrumMasters.

As in a development team's sprint planning meeting, the community then estimated each item and decided they could commit to completing these tasks during the sprint. Two weeks later at its sprint review, this team showed its product owner, a member of the ETC, a list of local user groups and a plan to work with one of them twice a year, sharing the expenses of bringing nationally known speakers into the area.

- ☐ Add to your improvement backlog by looking at the section headings of the chapters in this book. Many of them were written with this possibility specifically in mind.
- ☐ Review any notes available from recent sprint retrospectives. These are often an excellent source of improvement backlog items.

**THINGS TO
TRY NOW**



**76 | Chapter 4** Iterating Toward Agility**Focus on Goals with Practical Relevance**

For an improvement community to have the most impact, its members must focus on goals of immediate and practical relevance to the development teams using or attempting to get started with Scrum. The best way to do this is for improvement community members to work side by side with development team members on something important to the development team. This is what Google's "test mercenaries" do. Test mercenaries are members of the testing community who are experienced engineers with a passion for and expertise in testing. They spend up to 20% of their time for three months on a project other than their own. During this time they add tests and refactor code as a direct help to the development team.

I suppose that test mercenaries could instead spend this time creating presentations and spreading the gospel of developer testing. Something tells me, though, they are better able to achieve their goals by working with a team rather than preaching to it. A development team that has had the help of a test mercenary ends up with improved code and more tests. It also witnesses the benefits of an additional focus on developer testing. This works wonders in motivating those on the Scrum development team to continue the effort after the mercenary moves on to another team.

Focusing on providing practical assistance to development teams also helps keep improvement community members from falling into the habit of preaching to the development teams. A common problem when adopting Scrum is that the early adopters often become zealots anxious to convert everyone else. What zealots often forget is that it took them time to get comfortable with the idea of Scrum and the changes it requires. When others fail to convert instantly, zealots often perceive the delay as resistance. Because zealotry and pushing others to rapidly adopt new ideas can cause more harm than good, it is important for improvement community members to understand that their role is to consult rather than preach (Allen-Meyer 2000c, 25).

Improvement Community Members

Organizational change expert Glenn Allen-Meyer says that change should be done "with, not to, the people expected to change" (2000b). Because of this, it is important that anyone with a passion for an improvement opportunity be encouraged to participate in its community. Membership should not, for example, be restricted to only the organization's most senior employees. Broad participation in improvement communities helps everyone in the organization feel that change is occurring with them rather than to them. There should be no limit on the number of people participating in an improvement community. Communities often include well over 100 members, with individual participation levels going up and down over time based on the other demands of each person's job.





Participating in a community is not meant to be a full-time job; it is something someone takes on in addition to regular work. Improvement community leaders at IBM are asked to contribute two hours per week, although many contribute more based on a desire to see more rapid progress. A participant's manager, product owner, and ScrumMaster, though, are responsible for ensuring that those passionate enough about a change to work toward it are given sufficient time to do so. Google accomplishes this by telling each employee to spend 20% of each week on something of interest. The time could be spent, for example, exploring a new product idea or participating in a community.

Successful Scrum adopter Salesforce.com has a similarly innovative approach it calls PTON, pronounced pee-tee-on and meaning "paid time on." Patterned after the common PTO ("pee-tee-oh") policy for paid time *off* in many companies, Salesforce.com's PTON program gives employees dedicated time at work to pursue initiatives of their own choosing. Each employee is given one week of PTON for each year with the company. Salesforce.com employees can use the PTON time to work on a community initiative, explore new product ideas, or do just about anything they want.

Google's 20% policy and Salesforce.com's PTON programs were not created specifically to allow people to work in an improvement community. And organizations do not need to make such dramatic changes just to get started adopting Scrum. An easy starting point is simply for managers to commit to freeing up some number of hours each week for those who want to work on an improvement community.

"We've been working on this new product for a year. We ship it in four weeks and, as the product owner, I need the team's full time and attention for the next four weeks."

OBJECTION

Absolutely. Team members probably already know this and have plans to scale back participation in any communities to the minimum possible over that period. A team member who generally feels valued and allowed to devote time to the longer-term initiatives of a community will willingly minimize community participation during a true crunch period because she knows she can devote more time to it later.





OBJECTION

"These improvement communities seem just like the Software Engineering Process Groups (SEPGs) our company created to push CMMI. Isn't this just a new name for an old idea?"

Not really, but I can understand why you might think so. Both ICs and SEPGs are focused on helping the organization improve how people develop software. However, while their goals are the same, an SEPG and an IC differ in a few subtle but important ways:

- An SEPG looks at the process and answers the question, "What could we improve?" Members of an improvement community look at their own projects and ask, "What could we improve?" and "What are we doing well that others should know about?"
- Some SEPGs force compliance with a process; an improvement community has no authority from which to force compliance.
- Some SEPGs are chartered to look only at portions of the overall development process. ICs are encouraged to look beyond the product development process to find improvement opportunities.
- Improvement communities are self-motivated and self-organizing. In general, no one is told to join an improved community. (Although this may occasionally be done to start a new community.)
- Members of an IC are more likely to take an experimental, try-it-and-see approach to process improvement.
- Improvement communities are ad hoc and organic, formed whenever passion for a topic brings people together. SEPGs are formally created and often discouraged from functioning in an ad hoc manner.

Disbanding a Community

Most communities will eventually disband. A community formed to promote automated testing, for example, may exist with members coming and going for years as long as that is an area in which the organization needs to improve. Eventually (at least we'd like to think), the organization becomes good enough at automated testing that those community members can contribute more by devoting time to other improvement communities and the opportunities they represent.

Regarding the ETC specifically: It should disband once the organization has realized its transition to Scrum and has entered a phase of continuous improvement. The ETC exists only during the transition period, which may be multiple years for a large transition.





- ❑ Identify an improvement you are passionate about. Ask two or three coworkers to help you. Create an improvement backlog and plan a first sprint. Even if you can manage only an hour a week on it, start. As you begin to make progress, incorporate your improvement in the work of your team or offer it to another team. Generate interest by telling (or, even better, showing) others what you've accomplished.

**THINGS TO
TRY NOW**

One Size Does Not Fit All

In this chapter, I've presented a community-driven approach to Scrum adoption. A guiding community—the Enterprise Transition Community—does some of the work of the transition, but most important it creates an environment that encourages other communities to form. These communities—called improvement communities—are formed when a group of employees choose to work together to improve the organization's use of Scrum. Both types of communities use Scrum to drive the organization toward becoming agile.

But one size clearly does not fit all. The approach I am describing in this chapter works well when transitioning a medium or large department to Scrum. Scale it down as appropriate. A software department of 20 professionals, for example, may benefit from having one group of passionately agile individuals who help drive change and improvement. They are both an ETC and an IC in that case.

Looking Forward

So far, in the chapters that make up the initial section of the book, we've discussed why transitioning to Scrum is hard, but worth it. We've talked about the activities that accompany change and some tools you can use to help people make the switch to Scrum. We've discussed patterns for adoption that can guide our general approach to transitioning to Scrum. Finally, we've looked at how to combine all of that information, and the Scrum process itself, and use it to manage a Scrum adoption, on any scale. Throughout the first four chapters, I've made a point of saying that, unlike other change initiatives, with Scrum there is no end state. There is no point when you're done. Instead, Scrum requires continuous improvement, which can be managed through improvement communities, using Scrum itself.

In our next chapter, we discuss how to pick your first project, your first team, and get started with the business of becoming agile with Scrum.





80 | Chapter 4 Iterating Toward Agility

Additional Reading

Conner, Daryl R. 1993. *Managing at the speed of change: How resilient managers succeed and prosper where others fail*. Random House.

In this book, Conner describes eight key patterns of how people behave during organizational change. One of the goals of his process for change management is to foster resilience in people and organizations. His view of resilience is compatible with this book's presentation of change as continuous and agility as something to be iterated toward.

Katzenbach, Jon. R. 1997. *Real change leaders: How you can create growth and high performance at your company*. Three Rivers Press.

Katzenbach's book is based on extensive interviews with individuals who he found to be the true source of change in organizations. These are the "real change leaders" of the book's title. The book contains many engaging stories about individuals who would make good improvement community members.

Kotter, John P. 1996. *Leading change*. Harvard Business School Press.

Kotter's highly respected book is a classic on organizational change. In it, he lays out an eight-step process for creating change. In his second step, Kotter advocates the creation of a guiding coalition, which has some similarities to an ETC. Additionally, his article in *Harvard Business Review* (1995) offers a concise summary of this book.

Schwaber, Ken. 2007. *The enterprise and scrum*. Microsoft Press.

In this book, Schwaber, the coinventor of Scrum, describes what is necessary to transition an entire organization to Scrum. Included is advice on the improvement backlog and on the Enterprise Transition team, which is similar to the Enterprise Transition Community as I have presented it.

Wenger, Etienne, Richard McDermott, and William M. Snyder. 2002. *Cultivating communities of practice*. Harvard Business School Press.

Wenger is recognized as the authority on communities of practice. This highly readable book describes everything you need to know to begin cultivating communities within your organization, including a chapter dedicated to advice to community coordinators.

Woodward, E.V., R. Bowers, V. Thio, K. Johnson, M. Srihari, and C. J. Bracht. Forthcoming. Agile methods for software practice transformation. *IBM Journal of Research and Development* 54 (2).

Members of IBM's Quality Software Engineering organization are using an approach very similar to the one described in this chapter to spread agile throughout IBM. This excellent paper describes how they function as an Enterprise Transition Community, ways in which they encourage improvement communities to form, and how they use the Scrum framework to drive improvements in how they use Scrum.

