

# CHAPTER 12

---

## The Product Coordination Team

*“In theory, theory and practice are the same. In practice, they are different.”*  
—Attributed to many

---

---

### IN THIS CHAPTER

This chapter describes the challenge of teams coordinating with each other. Self-interest lies at the root of this problem: Teams (rightly) focus on their immediate needs and work. The Scrum-of-Scrums is one approach that many organizations try. It can work when teams have natural affinity but when there are competing needs, it is not effective. The Product Coordination Team (PCT) is a better alternative that works in all situations. In a sense, the PCT acts as a product champion for cross-team coordination issues.

### Takeaways

Key insights to take away from this chapter include

- The Scrum-of-Scrums approach does not work well when teams do not have common motivations or interests to bind them.
- The Product Coordination Team is chartered to focus on cross-team issues with the authority to place stories in individual teams’ product backlogs when needed.
- PCT membership comprises both permanent and rotating members.
- The work of the PCT continues across the iteration cycles.

## Getting Teams to Work Together

Many organizations attempting to scale Scrum to the enterprise ask how to get teams to work together. The standard approach is to use a Scrum technique called the “Scrum-of-Scrums.” The notion of Scrum-of-Scrums is to have members of different teams meet regularly to foster collaboration, communication, and shared requirements. While we have used this technique, and seen others use it successfully to coordinate teams that were closely related, more often we have seen companies struggle with it when they were attempting to coordinate teams across an organization or teams that have disparate goals. In other words, if you have a large-project team with a common goal, Scrum-of-Scrums may work well; however, if you have several project teams with different goals, we believe the approach is fundamentally flawed.

### ***Scrum-of-Scrums***

The Scrum-of-Scrums is a coordination team that is composed of a representative (usually a technical person) from each team. They gather on a regular basis to discuss issues of collaboration across the teams. These meetings are typically held weekly but may be held more often as required. A typical agenda for these meetings is:

1. During the first 15 minutes, each participant answers four questions:
  - a. What has your team done since we last met?
  - b. What will your team do before we meet again?
  - c. Is anything slowing down your team or in its way?
  - d. Are you about to put anything in another team’s way?
2. The rest of the meeting is spent resolving problems and discussing issues on the teams’ backlogs.

### **Factors that Work against Scrum-of-Scrums**

What works against the Scrum-of-Scrums? There are three factors: team perspective, team motivation, and human nature.<sup>1</sup> Teams usually take a

---

1. We are very much pragmatists. Theoretically, there is no reason that Scrum-of-Scrums can’t work in many of the places it is attempted. But we are more concerned with what happens when teams actually use it. These factors get in the way enough of the time to make it a poor practice.

local perspective—their work is somehow more important or otherwise different from the work of the other teams. A common expression is “can’t see the forest for the trees.” Certainly, they are motivated to focus on their own work because most companies measure and reward people based on the work they do directly or, possibly, based on the success of their own projects. Rarely are rewards based on how their department—or how the company as a whole—operates. This tends to focus them on their own needs, not on the needs of other teams that may need their help. Thus, when a Scrum-of-Scrums is just about sharing information, it may work well because there is no conflict of interest. But when it comes to coordinating work—when one team has to help another at the expense of its own work—the Scrum-of-Scrums does not work.

Why can’t teams overcome this bias? Why don’t team members recognize that when they help each other, they are impacting the company as a whole to do better? Why can’t they see the big picture? Is it mostly the fault of how they are measured? Very often, but that is not the only reason. People tend to identify more closely with the people they work directly with than they do with others in the company. This is readily observable. If you have ever been a member of a large organization, you can easily see this. People are most interested in how their immediate coworkers are doing, then how their department is doing, then their division, and finally their company. This is just human nature.

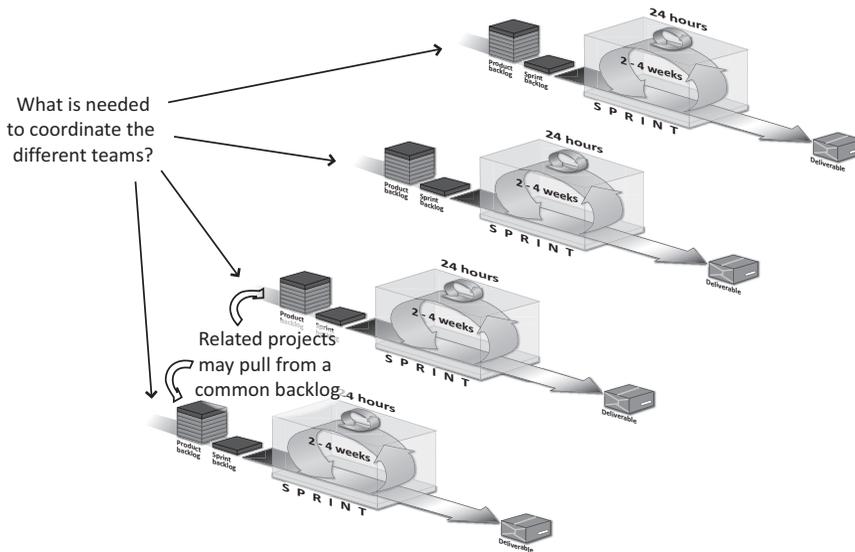
Scrum-of-Scrums fights against this basic trait. It assumes that putting individuals from various teams together in a loose confederation creates a bigger view of the organization across all of the teams. There is little evidence that this works in practice, particularly when self-interest is involved.

A different approach is usually needed. A team comprised of members with the same perspectives, motivations and purposes must be formed. You should not expect a person to be effective when they are drawn by two sets of loyalties. People just don’t work that way.

### ***The Challenge of Coordinating Teams***

Consider the four teams in Figure 12.1. The two teams at the bottom of the diagram are pulling from exactly the same product backlog. While they might share some common interests, as a whole the four teams won’t have the same vested interest in the bigger picture. For example, let’s consider how multiple teams working together may view these issues:

- Progress across teams



**Figure 12.1** Several teams working together

- Requirements that need multiple teams to implement them
- Technical dependencies between teams
- Common components that are used by several teams
- The need for one team to modify their code to assist another team
- Code shared by the teams
- Knowledge one team has that another team needs

Their views on these issues are addressed as follows:

- **Progress across teams.** Management needs to see the progress that teams are making as a whole. Consolidated burn-down and burn-up charts present most of this needed information. However, management often looks for a more comprehensive and aligned view of the entire operation. A meeting of teams can be a good approach for this combined view. A Scrum-of-Scrums works well here because no conflict exists among the team members.
- **Requirements that involve multiple teams to implement them.** Very often, multiple teams are required in order to imple-

ment a particular story. For example, consider a development organization that has a database team. Stories requiring changes to the database may require both the team writing the story and the team responsible for maintaining the database. A Scrum-of-Scrums often works well here because the teams are coming together to solve a shared problem.

- **Technical dependencies between teams.** Frequently, teams use software that is developed by other teams. The APIs for these service classes need to be discussed by both teams. The teams need to coordinate with each other while the modules are being developed. Teams need to warn one another when they are about to change a component or function that another group needs to use. This avoids what we call “clobberation.”<sup>2</sup> The Scrum-of-Scrums approach has a mixed track record here. If the team providing the code to be used functions as a service organization to the teams that are using their code, Scrum-of-Scrums works well because it is a good method for both groups to get their jobs done. However, a conflict of interest may arise if what one team wants adversely impacts the team that is writing the code. The Scrum-of-Scrums does not help resolve such conflicts.
- **Common components that are used by several teams.** Large organizations require common components for several different teams. Some sort of collaboration to identify and define these is necessary. When a component team is set up to meet this need, Scrum-of-Scrums can be a good way to communicate between the component team and the others since it helps all of them get their jobs done.
- **The need for one team to modify their code to assist another team.** Teams often build software that is similar, if not exact duplicates. Collaboration between teams is necessary to avoid duplication. This is particularly important in Agile environments because software evolves over time. This means teams must collaborate as software is developed so that they notice when duplication (or almost duplication) is being produced. The challenge comes when

---

2. “Clobberation” is a term coined by Alan Shalloway to describe the effect one team has on another team with which it is collaborating when the first team does not communicate changes it is making that will adversely affect the second team.

one team (“Team A”) writes something that another team (“Team B”) could use if only Team A’s code were modified just a little. If Team A is too busy with other things, Team B often resorts to copying and pasting the code or rewriting the code completely. Scrum-of-Scrums may find this a potential case for re-use, but even if it does, there is no guarantee that the people on Team A will modify the code since they may be under pressure to do their own work.

- **Code shared by the teams.** Sometimes duplication happens when one team builds something and another team could use a variant of it. If an up-front design approach were being used, a common interface might be settled on early. But with emergent design, these interfaces must be discovered when the second variant is needed. Collaboration among different teams is necessary to see this. This results in the same situation as the previous case: One group needs to update their code for another’s use.
- **One team has knowledge that another team needs.** Very often, one team needs knowledge that another team has. Unfortunately, they do not always know which information it is. Frequent communication across teams can assist with this. While Scrum-of-Scrums may work here, it is often hit or miss.

In theory, Scrum-of-Scrums should work because it provides teams a way to communicate with each other and all members want overall success. But the reality is that Scrum-of-Scrums does not create a bigger view and requires people to step outside their immediate concerns and look at the bigger picture—one they may not agree with across the teams. Something more is needed.

## The Product Coordination Team

In all of this, the root cause for teams not collaborating is that their perspective is too narrow, focused on their own local needs. To achieve collaboration, they need a perspective that works with their self-interest.

What is needed is a coordination structure that is fundamentally focused on the larger organization’s perspective. Rather than a loose confederation, like the Scrum-of-Scrums, it is chartered to look more broadly: It identifies and prioritizes stories involving cross-team issues and assigns stories to other teams to do the work when it is most appropriate for them. We call this the “Product Coordination Team” (PCT).

In a sense, the Product Coordination Team becomes another product champion for the team, one that prioritizes those stories that involve not the product the teams are building, but the way teams are working together.

While there are many similarities between the PCT and the Scrum-of-Scrums, the dynamics are quite different. The PCT is focused on a higher view: “optimizing the whole.” The PCT is a true team, composed of members from different teams, unified to reach a common goal. Scrum-of-Scrums tends to devolve into individual teams representing their own team’s issues in the bigger picture.

---

### **Case Study: Product Coordination Team**

**GROUP PROFILE:** Web Applications for large healthcare insurance company

**CHALLENGES:** Multiple scrum teams unable to coordinate using Scrum-of-Scrums

**INSIGHT:** Multiple cross-functional teams were having coordination challenges. As they pulled work from merged backlogs, teams began to step on one another as they touched the same code baseline. A Product Coordination Team was created, using architects and technical leads. The multiple teams synchronized their iterations to start and stop on the same schedule. This enabled the teams to demonstrate, plan, and retrospect together. Putting the multiple teams together (with the PCT present), enabled the teams to present their upcoming iteration plans to each other. It was much easier to identify opportunities for encapsulating effort when the teams presented their committed stories. As the PCT identified these duplicate efforts, they created stories that were distributed to the appropriate teams so that each group could safely focus on implementing code with minimum overlap. This model also enabled the teams to recognize when they were potentially duplicating effort, so the PCT was able to guide effective abstraction so that duplicate code was avoided.

---

### ***Product Coordination Team Membership***

The PCT is composed of both permanent and rotating members.

- **Permanent Members.** Permanent members are useful to maintain consistency as well as to ensure the team’s purpose is understood.
- **Rotating Members.** Including rotating members from development teams helps keep the PCT from becoming too removed from the needs of the development teams; it also means it has members who readily identify with the development teams. The PCT is first and foremost a service team. Although it will be creating stories for the teams to work on, its purpose is really to improve how the

development teams function as a whole. These members stay for a certain number of iterations or one to two months.

- **Planning Members.** Planning members come from the development teams and participate on the PCT only during the iteration planning sessions.

### ***Product Coordination Team Guidelines***

First and foremost, the Product Coordination Team is about maximizing business value. This comes from:

- Determining and then coordinating how the company can maximize the technical synergy among the development teams. In particular, it is looking for shared components, avoiding redundancy, assisting emergent design, and avoiding “clobberation.”
- Validating elevation plans (see chapter 7, Lean-Agile Release Planning).

The activities of the PCT vary during the iteration. Tables 12.1 and 12.2 offer some basic guidelines for a Product Coordination Team.

**Table 12.1** Activities of the Product Coordination Team

<b>Time Period</b>	<b>Activities</b>
Day before iteration planning <i>or</i> Before the first planning day (if the teams being coordinated plan on different days)	<p>The PCT meets with the product champions of each team to discuss possible coordinating stories. This meeting is attended by all members of the PCT, including at least one person from each development team that doesn't have a permanent or rotating PCT member.</p> <p>Discuss the stories that each team needs to do. If any one of these stories is directly associated with a story already on the backlog, it can be made dependent upon that story's being done.</p> <p>Note that although the next iteration's stories are not certain, just before the iteration it is reasonably clear what the team will do next.</p>

**Table 12.1** Activities of the Product Coordination Team, **continued**

<b>Time Period</b>	<b>Activities</b>
During iteration planning	Each PCT member now goes to their corresponding team's planning session. At this point they present the stories the PCT has created to their associated team. New PCT stories may also be created at this time.
After iteration planning	The members of the PCT get together to review what has taken place and see if any new stories are needed or if any stories can be removed or deferred.
During the iterations	Permanent and rotating members meet with teams on an as needed basis to look ahead for future coordinating stories.

**Table 12.2** Guidelines for the PCT

<b>Topic</b>	<b>Guidelines</b>
Authority	The PCT does not have any command-and-control authority over the teams it is coordinating; however, they can place stories on any team's product backlog at any place on the backlog. In many ways the PCT and the product champions fill similar roles—both determine what needs to happen next. The PCT acts to prioritize the work required between teams while the product champion prioritizes work for a team.
Architecture	The PCT is not an architecture team, although it is often composed of many people from an architecture team if one exists. The PCT is not a vehicle for forcing particular design approaches onto teams.
Over-committing teams	The PCT can only place stories on teams' product backlogs. They cannot tell the teams they have to do "just one more story."
Doing the work themselves	The PCT typically should not do the work themselves. Instead, they write stories that one team needs another team to do.

### ***Mentoring***

The PCT can also provide the framework for a mentoring organization. Typically, the people on the PCT are of personality types that would make good mentors. They can also become the basis for a community of practice and other knowledge management/sharing methods.

### **Summary**

This chapter describes the challenge of teams coordinating with each other. Self-interest lies at the root of this problem: Teams (rightly) focus on their immediate needs and work. It is hard for them to look at larger issues. This is not a bad thing, it is a human thing. The product coordinating team (PCT) is a true team-of-teams that has the proper perspective and charter to focus on cross-team issues. Membership involves both permanent and rotating positions. In a sense, the PCT becomes a product champion for cross-team coordination issues. While it is similar to a Scrum-of-Scrums, the dynamics are very different.

### **Try This**

These exercises are best done as a conversation with someone in your organization. After each exercise, ask each other if there are any actions either of you can take to improve your situation.

- What are some challenges you have seen in getting teams to coordinate with each other?
- Who would be likely candidates for a PCT for our projects?
- Discuss how metrics in your organization assist team collaboration or hurt it.