Chapter 15. Multilevel Planning

On Scrum projects, we plan at multiple levels of detail and at multiple times throughout product development. In this chapter I provide a high-level, top-down description of the various Scrum planning activities and how they are interrelated. In the next several chapters I will explore portfolio planning, product planning (envisioning), release planning, and sprint planning in greater detail.

Overview

When developing a product with Scrum, planning takes place at multiple levels (see Figure 15.1).

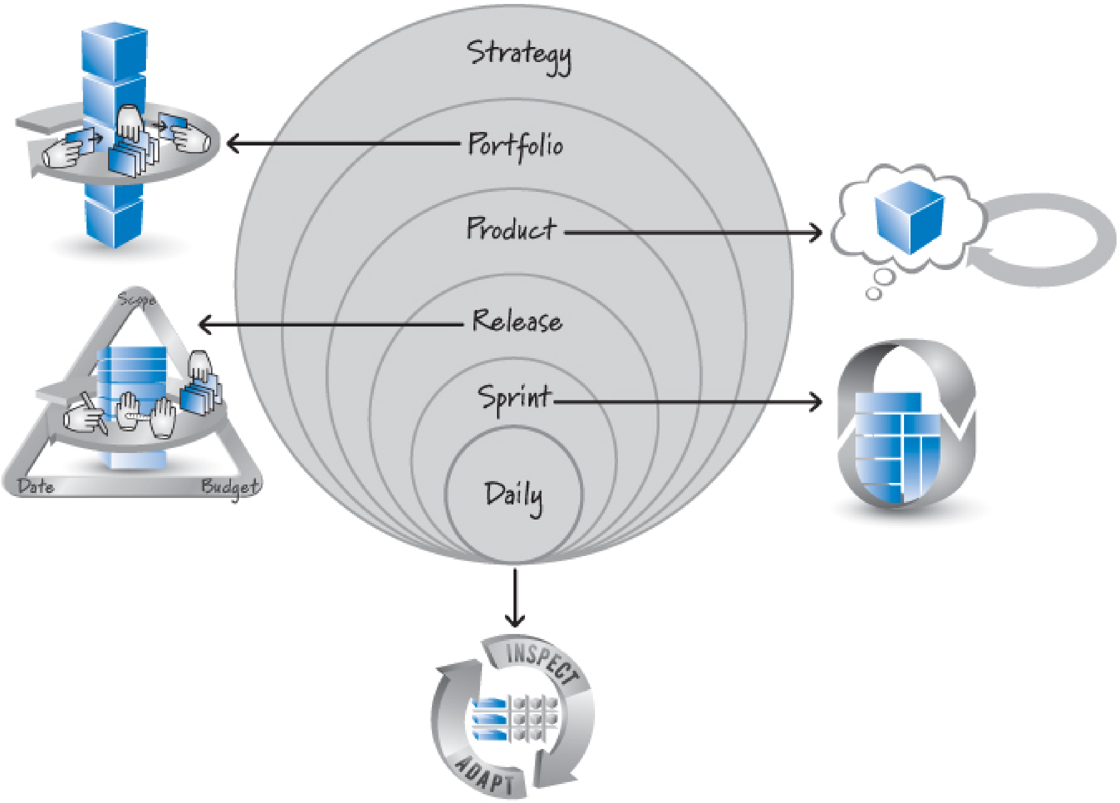


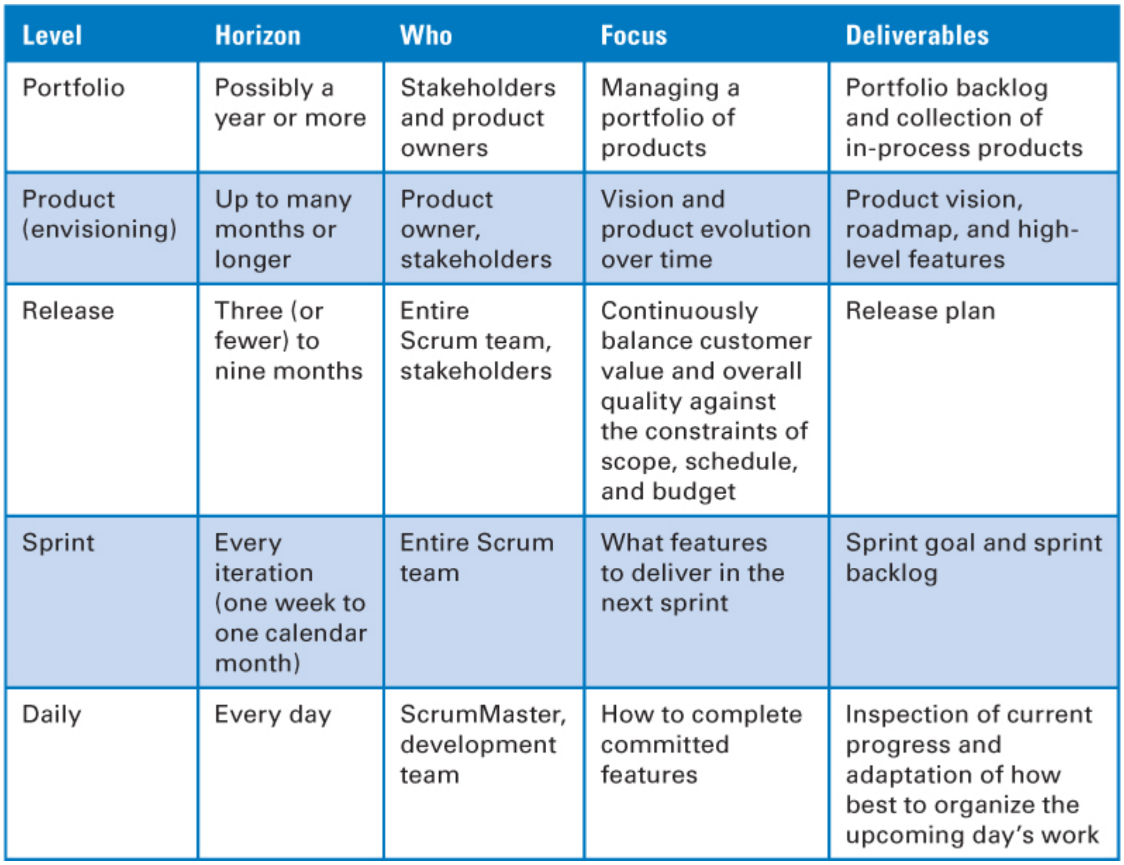
Figure 15.1. Different levels of planning

At the highest level is strategy planning, which although critical to an organization's success is outside the scope of this book. Formally Scrum defines only sprint planning and daily planning (via the daily scrum).

However, most organizations will benefit from portfolio, product, and release planning, so I will summarize approaches for each of them in this chapter and then discuss each in detail in subsequent chapters.

Table 15.1 summarizes five types of planning, with an emphasis on the span of time typically covered by each type, who is involved, what the focus is, and what the deliverables are at each level.

Table 15.1. Planning Level Details



Level	Horizon	Who	Focus	Deliverables
Portfolio	Possibly a year or more	Stakeholders and product owners	Managing a portfolio of products	Portfolio backlog and collection of in-process products
Product (envisioning)	Up to many months or longer	Product owner, stakeholders	Vision and product evolution over time	Product vision, roadmap, and high-level features
Release	Three (or fewer) to nine months	Entire Scrum team, stakeholders	Continuously balance customer value and overall quality against the constraints of scope, schedule, and budget	Release plan
Sprint	Every iteration (one week to one calendar month)	Entire Scrum team	What features to deliver in the next sprint	Sprint goal and sprint backlog
Daily	Every day	ScrumMaster, development team	How to complete committed features	Inspection of current progress and adaptation of how best to organize the upcoming day's work

To illustrate planning at each of these levels, I will use the redesign of the Scrum Alliance website (www.scrumalliance.org) as an example. The relevant context for this product is that in 2006 the Scrum Alliance, a nonprofit organization focused on the worldwide promotion of Scrum, had a dreadful website. It wasn't pretty, was hard to navigate, and was content poor. When I became the Managing Director of the Scrum Alliance at the end of 2006, one of the first things the board of directors asked for was a new and much better website. I was the product owner for this effort and will describe the hierarchy of planning we performed to realize a new website.

Portfolio Planning

Portfolio planning (or portfolio management) is an activity for determining which products to work on, in what order, and for how long. Although portfolio planning is conceptually at a higher level than product planning (because it deals with a collection of products), one of its primary inputs is a newly envisioned product idea from product planning.

In 2006, the Scrum Alliance was a relatively new organization, and its portfolio contained only the ongoing development of its existing website. Once the initial envisioning of the new Scrum Alliance website was completed, the board of directors (the stakeholders of the Scrum Alliance portfolio backlog) approved the development of the first release of the new website.

Product Planning (Envisioning)

The goals of product-level planning (which I also refer to as **envisioning**) are to capture the essence of a potential product and to create a rough plan for the creation of that product. Envisioning begins with the creation of a vision, followed by the creation of a high-level product backlog and frequently a product roadmap.

Vision

The **product vision** provides a clear description of the areas in which the stakeholders, such as users and customers, get value. In our case, the users were the 10,000 worldwide members of the Scrum Alliance at the time (at the end of 2011 there were 150,000 worldwide members). The customer, who paid for the new product, was the Scrum Alliance board of directors on behalf of the members.

Our vision for the new Scrum Alliance website was as follows:

> For people worldwide who are interested in Scrum, the new Scrum Alliance website will be their trusted source of Scrum knowledge. It will be feature and content rich and will be their first stop on the Internet for learning more about Scrum or to collaborate on Scrum topics of interest.

High-Level Product Backlog

Once a product vision has been established, the next step is to generate an initial high-level version of the product backlog. In the case of the re-designed Scrum Alliance website, at the end of 2006 we already had a growing product backlog of features that the stakeholders and users wanted for the new and improved website.

Product backlog items included the following epic-level user stories:

> As a Certified Scrum Trainer I want to be able to post my public Scrum class on the Scrum Alliance website so that the community will know the details surrounding where and when I am offering the class.

> As a prospective student I want to be able to see details of all publicly available Scrum classes so that I can find one that meets my criteria for attendance.

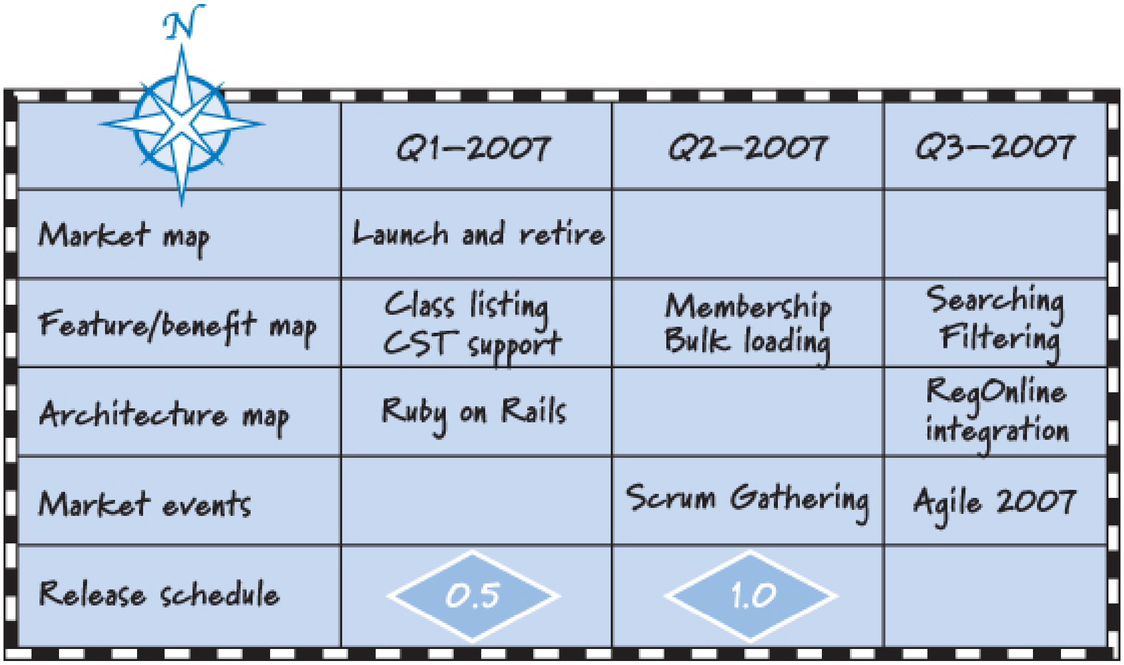
If our product had been completely new, we would have had to do at least some minimal up-front requirements generation to populate our product backlog and estimate at least the highest-priority items. In our case we had some product backlog items that we used as a starting point for ideas to be included in our vision of the new website.

Product Roadmap

Once a product vision and high-level product backlog have been established, it is helpful to build a product roadmap (sometimes referred to as a release roadmap). A **product roadmap** communicates the incremental nature of how the product will be built and delivered over time, along with the important factors that drive each individual release.

Today many organizations are striving for **continuous deployment**, where they deploy working features into production as soon as they become available. If your organization is focused on this practice, you might not need to produce a product roadmap. However, even if you do intend to deploy continuously, a product roadmap might be a useful tool for helping your organization think about larger collections of features, constraints that might dictate which features should be done around the same time, and when certain features should be available.

Figure 15.2 shows a product roadmap in a format promoted by Luke Hohmann (Hohmann 2003).



	Q1–2007	Q2–2007	Q3–2007
Market map	Launch and retire		
Feature/benefit map	Class listing CST support	Membership Bulk loading	Searching Filtering
Architecture map	Ruby on Rails		RegOnline integration
Market events		Scrum Gathering	Agile 2007
Release schedule	0.5	1.0	

Figure 15.2. Scrum Alliance website product roadmap

Shown in the roadmap are two releases, one in each of the first two calendar quarters of 2007. The “0.5” release in Q1 2007 was the first release of the new website; we chose that number because we planned for this first release to have fewer than half of the features of the old Scrum Alliance website, but it would include new features that were better than those of the old website. The desired features centered on listing all publicly available Scrum classes anywhere in the world and basic support for Certified Scrum Trainers (CSTs). Release 0.5 was a **fixed-scope release** because we knew the specific features we wanted to have on the new website before we could retire the old site. What we didn't know was how long it would take to get those features ready for launch. In Chapter 18 I will discuss how to determine the ship date for a fixed-scope release.

Release 1.0 was a fixed-date release. We knew we wanted the release to coincide with a Scrum Alliance conference (called a Scrum Gathering) that began on May 7, 2007, in Portland, Oregon. Our goal was to have an exciting set of features available by the first day of that conference. What we didn't know was how many features we could get into that release. In Chapter 18 I will discuss how to determine the content of a fixed-date release.

To summarize, on the Scrum Alliance website initial product roadmap we identified both a fixed-scope release (0.5) and a fixed-date release (1.0).

No matter what product you are creating, by the end of product-level planning, you should have a product vision, a high-level product backlog populated with estimated user stories, and (optionally) a product roadmap. In addition, you might also produce other artifacts to provide decision makers with sufficient confidence to move forward to develop the product.

The outputs of product-level planning became inputs to portfolio planning, where the initial 0.5 release of the redesigned website was approved by the board of directors.

Release Planning

Release planning is about making scope, date, and budget trade-offs for incremental deliveries.

On most development efforts it is sensible and necessary to do initial release planning after envisioning (product planning) and before starting the first sprint associated with the release. At this point, you can create an initial **release plan** that balances how much you can develop in the release against when the release will be available.

To have some idea of what you can deliver by a fixed date or when you can deliver a fixed set of features, you need to create and estimate a sufficient number of product backlog items.

A simple way to visualize a release is to draw a line through the product backlog (see Figure 15.3). All of the items above the line are planned for the release, and all of the items below the line are not planned for the release. This line can move up or down in the product backlog as you gain better insight into the product. In Chapter 18 I will discuss how to determine the position of this line.

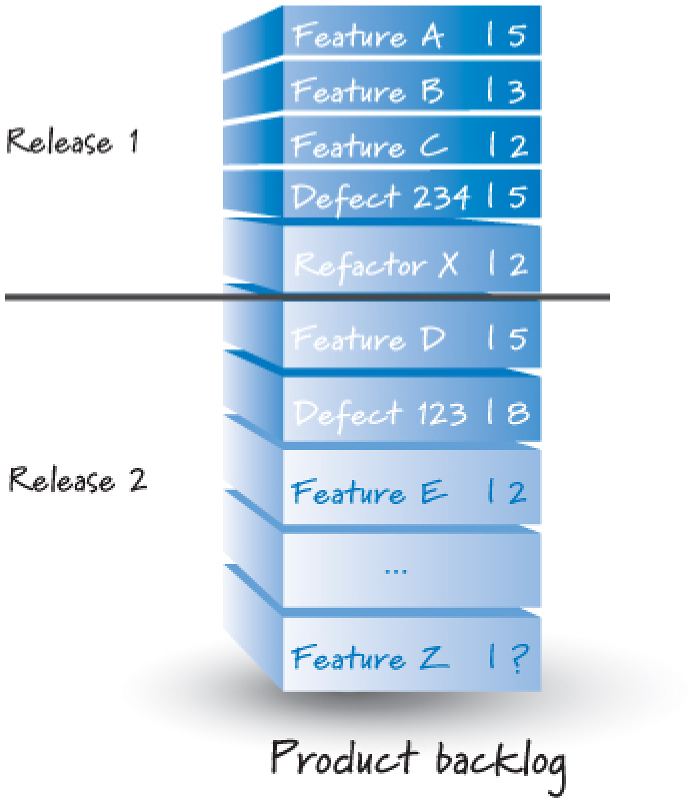
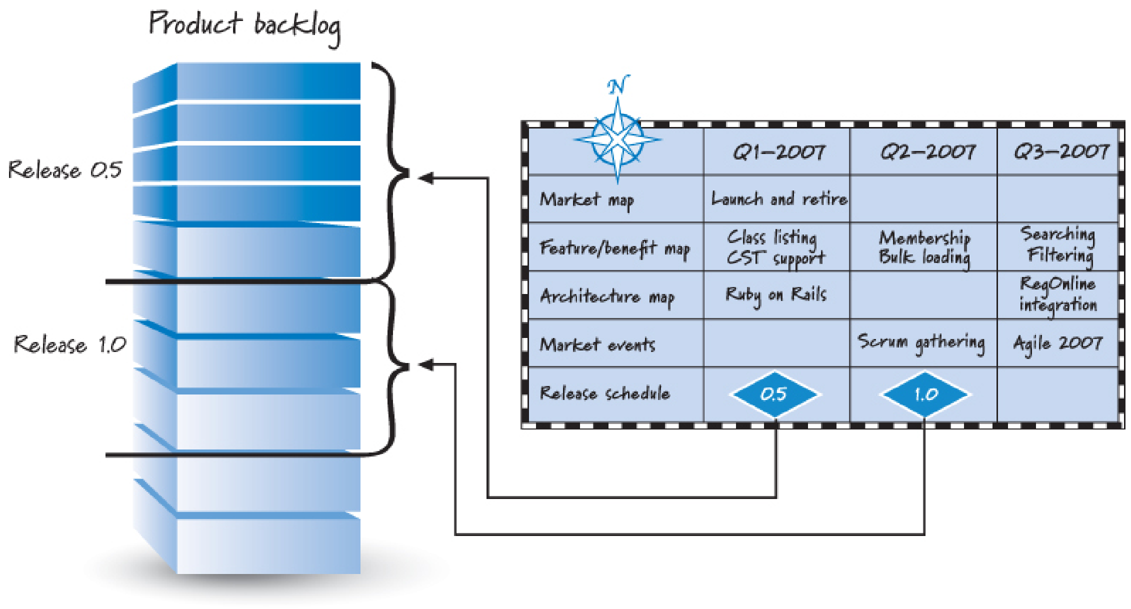


Figure 15.3. A release line in the product backlog

You can now easily tie the product roadmap to the product backlog to provide a more detailed elaboration of the contents of at least the near-term releases identified in the product roadmap (see Figure 15.4). A release on the product roadmap corresponds to a set of features in the product backlog.



	Q1–2007	Q2–2007	Q3–2007
Market map	Launch and retire		
Feature/benefit map	Class listing CST support	Membership Bulk loading	Searching Filtering
Architecture map	Ruby on Rails		RegOnline integration
Market events		Scrum gathering	Agile 2007
Release schedule	0.5	1.0	

Figure 15.4. Product roadmap releases mapped to the product backlog

The release plan must also have a time dimension associated with it, which can be expressed in terms of the number of sprints required to accomplish the release. Most releases are large and have more features than can be built in one sprint (see Figure 15.5).

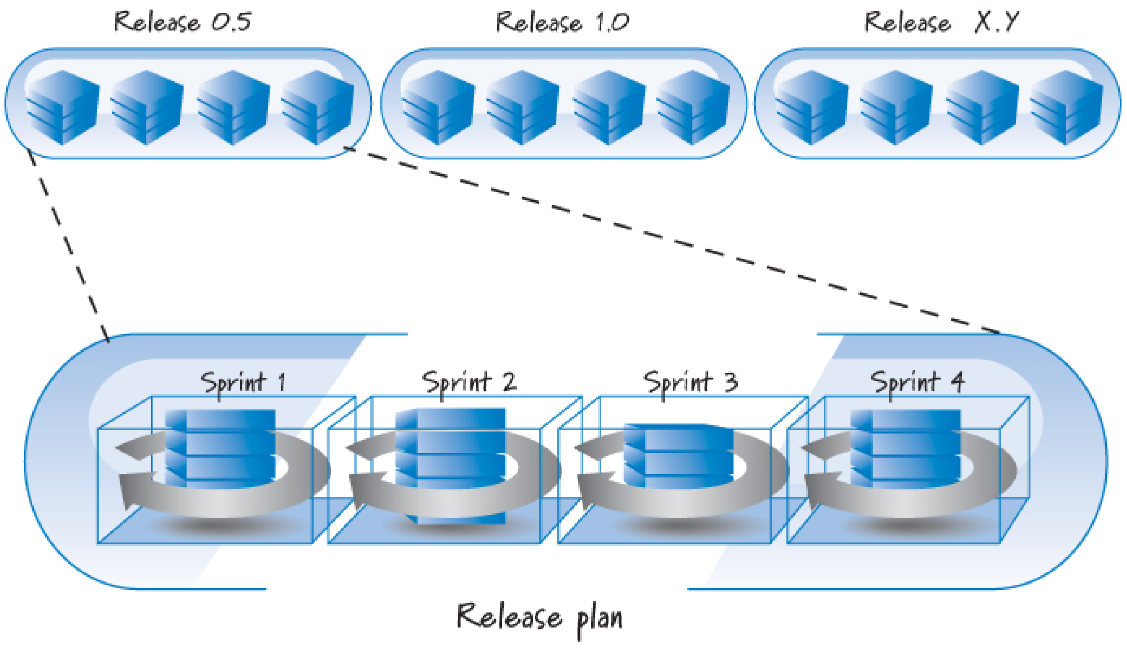


Figure 15.5. A release can encompass one or more sprints.

During release planning you might go so far as to guess the features that will be delivered in the first couple of sprints. This can be helpful when multiple teams need to coordinate work or when a team needs to request additional hardware, tools, or assistance in advance. But guessing ahead more than a couple of sprints is almost always unnecessary and violates the principle of doing planning just in time and just enough.

Sprint Planning

The specific product backlog items that the Scrum team will work on in the next sprint are agreed to at sprint planning, which occurs at the beginning of each sprint. During this activity, the team generates a sprint backlog: a description of the task-level work that has to be completed to get the product backlog items done (see Figure 15.6).

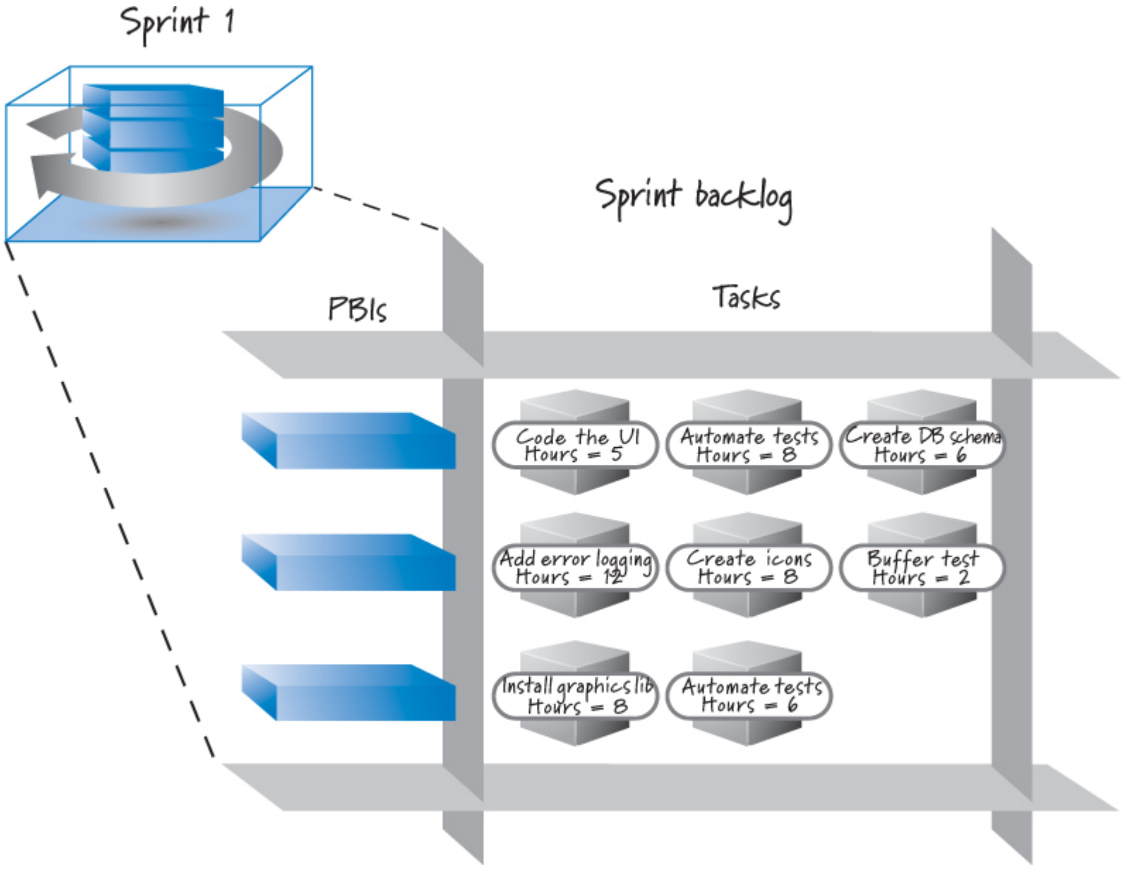


Figure 15.6. Each sprint has a sprint backlog.

During sprint planning the team does the next level of just-in-time detailed planning. I will discuss the details of sprint planning in Chapter 19.

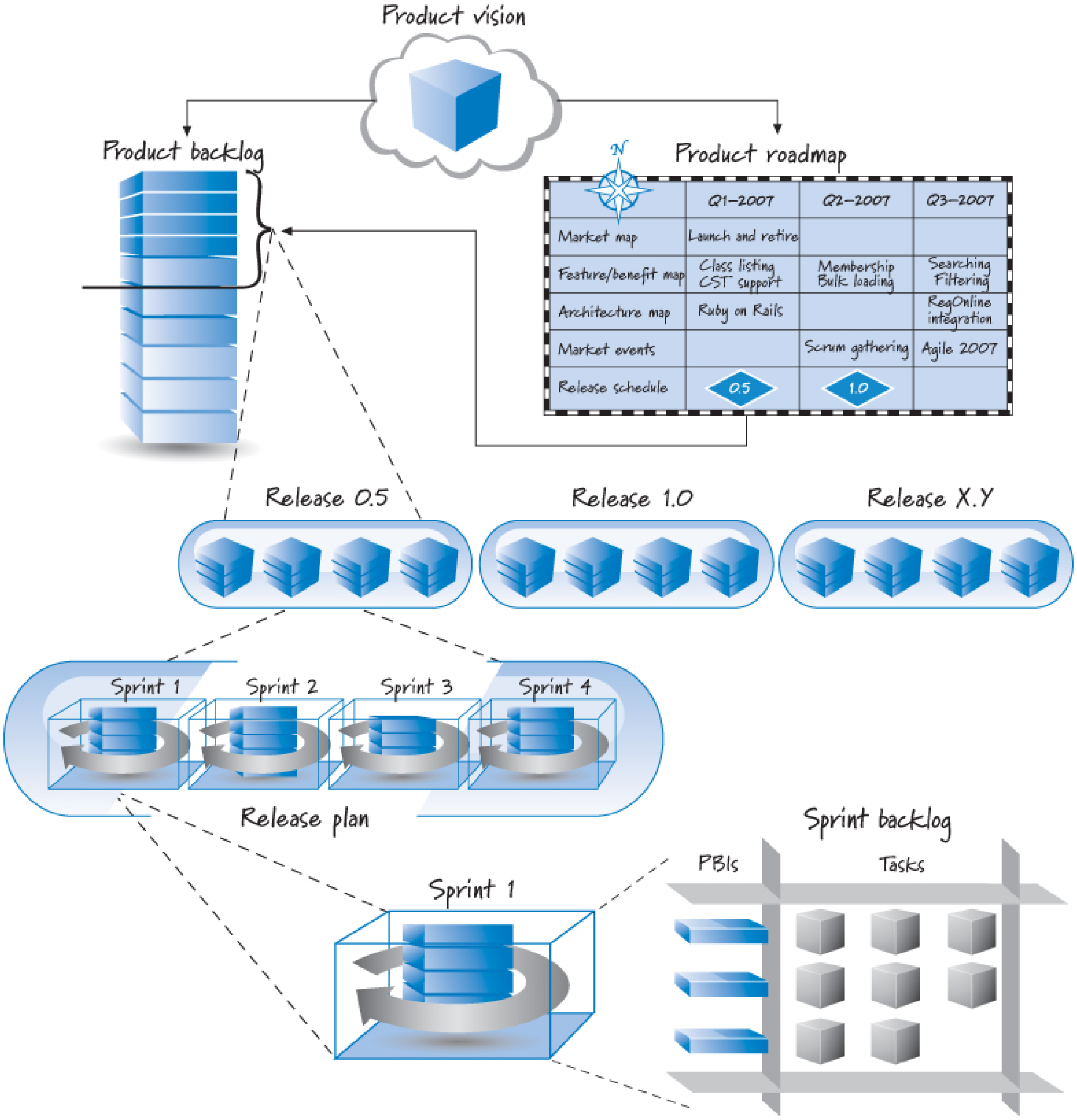
Daily Planning

The most detailed level of planning occurs during the team's daily scrum meeting. Recall that this is the activity where the team members get together and each person takes turns stating what she got done since the last daily scrum, what she is planning to work on today, and whether she has any impediments.

During the daily scrum, team members collectively describe, in a highly visible manner, the big-picture plan for that day. This also allows the team to use resource alerts. For example, someone might say, "Today I am going to work on the stored procedure task, and I should have that done by lunch. Whoever is going to work on the business logic task, please keep in mind that the business logic task is on the critical path for getting our work done this sprint and you should be ready to work on it right after lunch." Such communications can quickly identify potential work blockages and enable a better flow through sprint execution.

Closing

This chapter illustrated how planning at multiple levels of detail happens on a development effort using Scrum. Figure 15.7 (shown on the next page) graphically summarizes the artifacts produced at these levels (except the portfolio and daily planning levels) and their interrelated nature.



	Q1–2007	Q2–2007	Q3–2007
Market map	Launch and retire		
Feature/benefit map	Class listing CST support	Membership Bulk loading	Searching Filtering
Architecture map	Ruby on Rails		RegOnline integration
Market events		Scrum gathering	Agile 2007
Release schedule	0.5	1.0	

Figure 15.7. Hierarchical Scrum planning

In the next several chapters I will explore in greater depth the topics of portfolio planning, product planning, release planning, and sprint planning.