Pro .NET 2.0 Extreme Programming

Greg Pearman and James Goodwill

Apress®

Pro .NET 2.0 Extreme Programming

ISBN-13 (pbk): 978-1-59059-480-3

ISBN-10 (pbk): 1-59059-480-0

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Lead Editor: Dominic Shakeshaft
Technical Reviewer: Jason Lefebvre
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Jason Gilmore, Jonathan Gennick, Jonathan Hassell, James Huddleston, Chris Mills, Matthew Moodie, Dominic Shakeshaft, Jim Sumser, Keir Thomas, Matt Wade
Project Manager: Sofia Marchant
Copy Edit Manager: Nicole LeClerc
Copy Editor: Marilyn Smith
Assistant Production Director: Kari Brooks-Copony
Production Editor: Kelly Winquist
Compositor: Lynn L'Heureux
Proofreaders: April Eddy, Lori Bring
Indexer: Carol Burbo
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2560 Ninth Street, Suite 219, Berkeley, CA 94710. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

The information in this book is distributed on an "as is" basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is available to readers at http://www.apress.com in the Source Code section.

CHAPTER 11

Release Planning—The Journey Begins

In this chapter, we are going to practice what we have been preaching. We will start with the exploration phase of release planning by having the customer introduce his business problem. Then will have the customer create user stories based on the needs he identified in his problem introduction, starting with the highest priority features. When enough user stories have been created (more than will fit into a release), we will assign each user story a value (ideal days) estimate using story points. This will complete the exploration phase of release planning.

Then we will start the planning phase (planning game) portion of release planning. The customer will declare when he would like to have the first release delivered. The tracker will declare the release velocity of the team, based on the number of developers and the release date. The customer will then select a subset of all the user stories. These will be the focus of the release. They should not total more than the team's declared release velocity. The outcome of the entire release plan will be the scope of work that will deliver business value to the customer in the shortest amount of time possible.

Business Problem Introduction

Northwind Inc. is a large, independent reseller of unusual and exotic food and beverage products. To date, on the consumer side of the business, sales of Northwind products have been through direct-mail catalogs. People's orders are taken over the telephone, and then processed through Northwind's corporate systems. Numerous corporate systems process the order from initial input of the order all the way down to shipment and delivery of the order. Figure 11-1 shows the flow of an order through the Northwind corporate system.

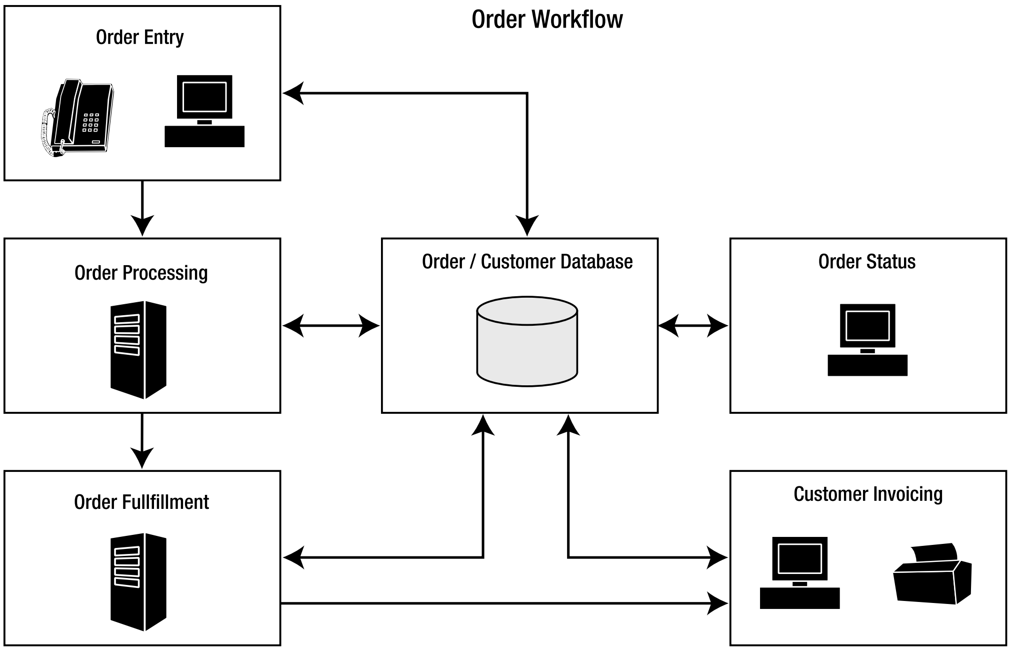


Figure 11-1. *Workflow of a customer order*

The Northwind senior management has defined the following mission statement for this project:

> *Create a Web presence that allows customers to self-order Northwind products and track the status of their orders all the way through to shipment.*

The objectives to be reached by this project over its life span are as follows:

- Put the entire product catalog on the Web with consumer access.
- Provide a secure means of allowing consumers to select and order Northwind products.
- Provide a secure means of allowing consumers to check the status of an order from order creation to order shipment.

Story Writing

Continuing with the exploration phase of our project, it's time for story writing. As you learned in Chapter 3, a user story contains one, and only one, business feature. It is written on an index card, has a title, and includes a few sentences that describe the feature.

Note Some customers may come to the release planning meeting with requirements already written out in one form or another. It is okay to refer to those requirements while creating the user stories, but the customer still needs to break up the requirements into user stories.

When we talked about writing this section, we discussed many different methods of conveying the information. We came to the conclusion that an example of the dialogue between the XP team and customer would best demonstrate the story writing process.

Helping the Customer Write the Stories

As we explained in Part 1 of this book, the customer is responsible for writing the stories. Of course, the customer gets some assistance from the XP team. The dialogue may go something like this:

XP team: Can you begin by providing the top features that you'd like to see in this release of the Northwind project? Try to keep these features simple and autonomous. We'll be writing each of these features down on index cards, where they'll become user stories. Remember, all we need at this point is a title and a few sentences that describe each defined feature.

Customer: The first requirement that must be part of this application is a login. The user must provide a valid user ID and password prior to doing anything else.

XP team: How do customers establish accounts with you?

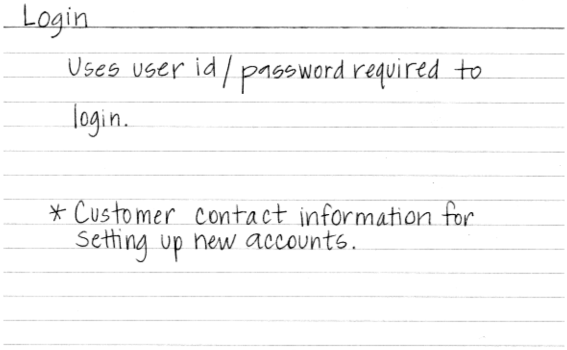
Customer: They have to contact us, and we'll create an account for them.

XP team: Should we include contact information on the login page?

Customer: Yeah, that's a good idea.

XP team: Okay, so why don't you title your first story something like Login and provide a couple of sentences describing the login functionality, including something about the contact information. Try not to be detailed at this point.

Customer: How does that look?



Login

Uses user id / password required to
login.

* Customer contact information for
Setting up new accounts.

XP team: Great. What's the next feature you're looking for?

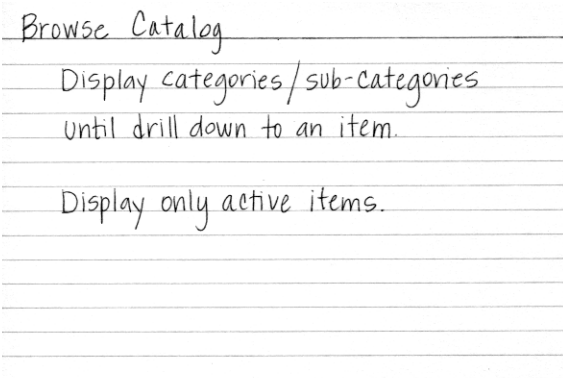
Customer: Once they log in, I'd like the user to be able to browse the items in my catalog.

XP team: Do you want to display all of the catalog items at once, or do you want to use some sort of category display?

Customer: I'd like to display my categories, and based on the category selected, display the items in that category.

XP team: Okay, now give this story a title and a brief description.

Customer: Is this descriptive enough?



Browse Catalog

Display categories / sub-categories until drill down to an item.

Display only active items.

XP team: That looks really good. I think you're getting the hang of this.

Customer: I just thought of something. My users are going to be browsing our system with Internet Explorer 5 and above. How do I capture this requirement?

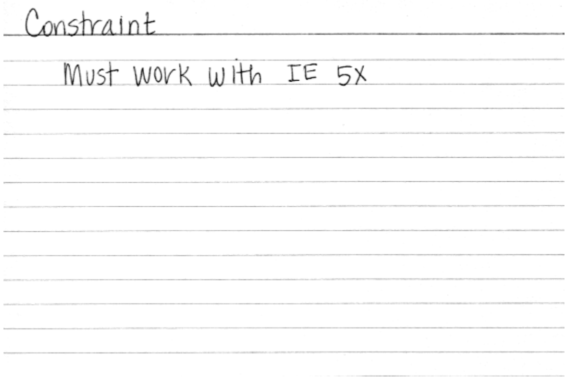
XP team: Well, something like this is really not a requirement—it's a constraint. In XP, we capture constraints much like user stories, except that they don't describe a feature, and we call them constraint cards.

Note A *constraint card* is just an index card that serves as a placeholder for constraint information, so the team and the customer do not forget about the constraint.

Customer: So, do I simply write the constraint on an index card?

XP team: Yes, you just want to make sure that you account for the constraint during the iteration phase of the project.

Customer: Is this good enough?



Constraint

Must work with IE 5X

XP team: That's great. Now, let's move on to your next feature.

Customer: I guess the next thing that I'd like to do is allow the user to perform a text search of the catalog.

XP team: What do you want to search on? Which attributes of the catalog item are you interested in?

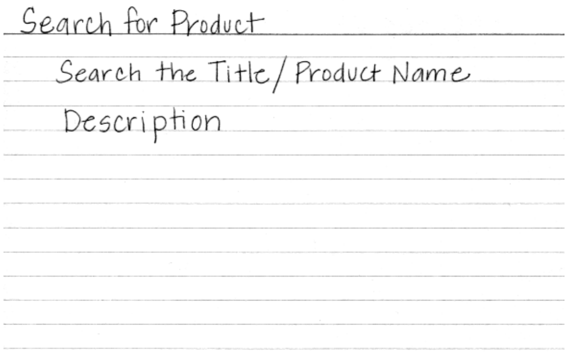
Customer: Well, at this point, I want to search on the item's title and description. What do we do if I decide later that I want to search on a different attribute of my item?

XP team: At this point, we'll only focus on your current need, but if you change your mind later, we'll create a new user story, and we may need to balance the release.

Customer: Okay, we can handle that later. For now, I'm only interested in the title and description.

XP team: Great. Go ahead and create your story card for the current search, and we'll move on.

Customer: Okay, here's the current story.



Search for Product

Search the Title/Product Name

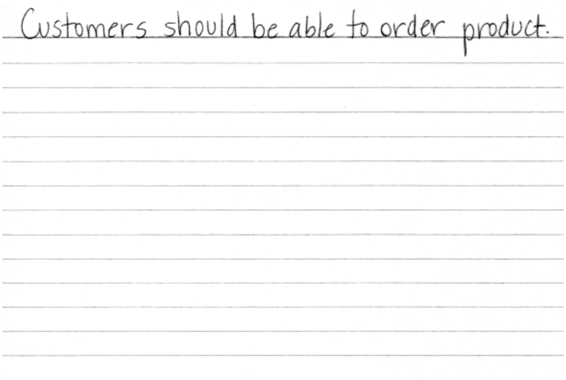
Description

XP team: What's your next feature?

Customer: I have a catalog. The user can browse the database. I guess I need a shopping cart. Do I just write down "shopping cart"?

XP team: That's more of an object. It really needs to be an action.

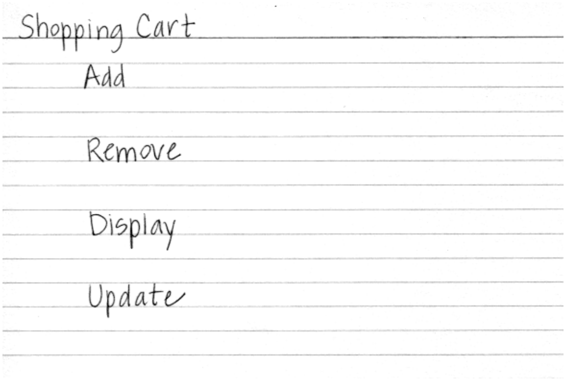
Customer: How about this?



Customers should be able to order product.

XP team: That looks more like a description. We may want to retitle this card. Go ahead and rip up this card and create a new one.

Customer: Okay, on this shopping cart, I'd like to add, remove, display, and update the shopping cart. How does this look?



Shopping Cart

Add

Remove

Display

Update

XP team: That looks better.

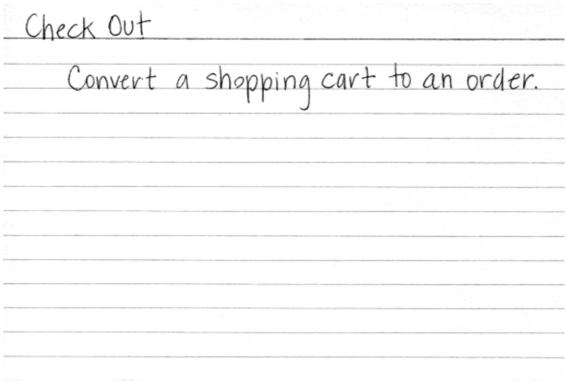
Customer: Next, I want to allow the users to order their selected items.

XP team: What do you want to call this story?

Customer: How about Check Out?

XP team: That sounds good. Why don't you provide a brief description?

Customer: What do you think of this?



Check Out

Convert a shopping cart to an order.

XP team: What does that mean other than transferring items from a shopping cart to an order? Do you need to gather any additional information?

Customer: No, we already have their account information. We may allow them to ship the product to a different address in the future, but for now, they can only use their account's shipping address.

XP team: Do they need an order confirmation?

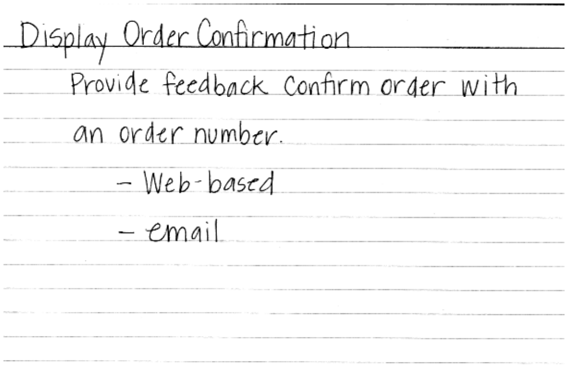
Customer: That's a good idea. What do you recommend?

XP team: Let's create a new story and call it something like Display Order Confirmation.

Customer: Okay, what kind of confirmation do you think we should use?

XP team: It's very common to provide an immediate order number in the response, and then also send an e-mail confirming the order.

Customer: That's an excellent idea. How does this look?



Display Order Confirmation

Provide feedback Confirm order with
an order number.
- Web-based
- email

XP team: Excellent. What's next?

Customer: I think the users should be able to look up an existing order's status. They should also be able to look at their order history.

XP team: It sounds like you've described two features—order status and order history. Why don't we break these features into two different stories?

Customer: Okay, I'll start with Display Order Status, and then describe another story named Display Order History.

XP team: Sounds good. What's Display Order Status?

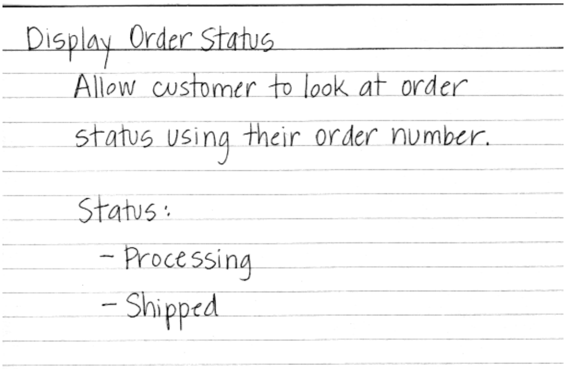
Customer: I think it should display the status of an order given an order number.

XP team: What are the possible statuses?

Customer: At this point, we only have two statuses—processing and shipped.

XP team: Okay, make sure we capture these values.

Customer: What do you think of this?



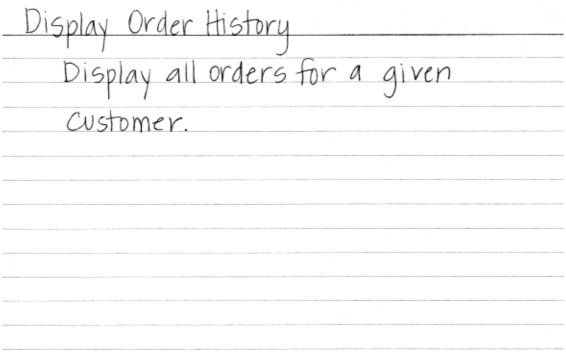
Display Order Status

Allow customer to look at order status using their order number.

Status:
- Processing
- Shipped

XP team: Fantastic! What about the Display Order History story?

Customer: That one is pretty easy. I just want the customer to be able to display the entire order history on a single page.



Display Order History

Display all orders for a given customer.

XP team: Outstanding. I think we're making great progress. What's next?

Customer: Let me think. At this point, I can browse, search, shop, create an order, check on an order, and recall all of a customer's orders. I think the next step is to add some administrative features.

XP team: What did you have in mind?

Customer: Well, I need to be able to add and edit customers. Should this be on the same card?

XP team: No, it really sounds like two functions—adding and editing.

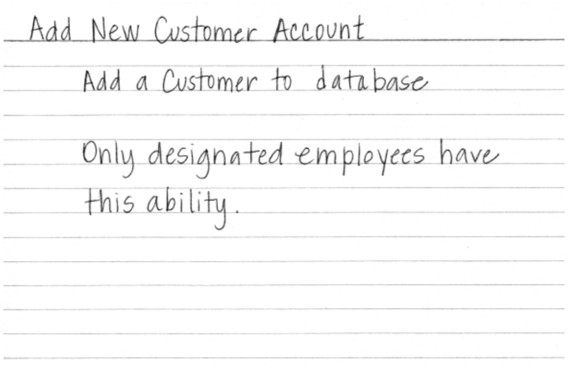
Customer: Okay, let's add a new customer first.

XP team: Before we do this, can anyone add a new customer or should they have a particular role?

Customer: No, only designated employees should be able to add new customers.

XP team: So, make sure you record that in your story.

Customer: How about this card, Add New Customer Account?



Add New Customer Account

Add a Customer to database

Only designated employees have this ability.

XP team: That's good. What about the customer edit?

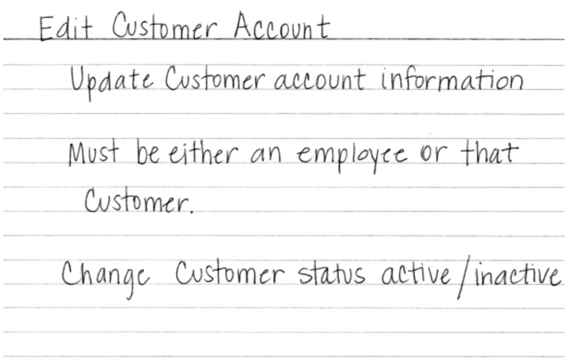
Customer: That should really be about the same.

XP team: Do you want to give customers the ability to modify their own accounts?

Customer: Yes, the only users allowed to edit a customer account are the customer or one of our employees.

XP team: Does a customer have a status? It might make sense to disable a customer account if he's delinquent. What do you think?

Customer: That makes a lot of sense. Why don't we give the customers a status of active or inactive? Does this capture enough information?



Edit Customer Account

Update Customer account information

Must be either an employee or that Customer.

Change Customer status active / inactive

XP team: Yeah, that's great.

Customer: I think I have one more feature to add. I want to add some product administration.

XP team: What kind of functionality do you think you'll need?

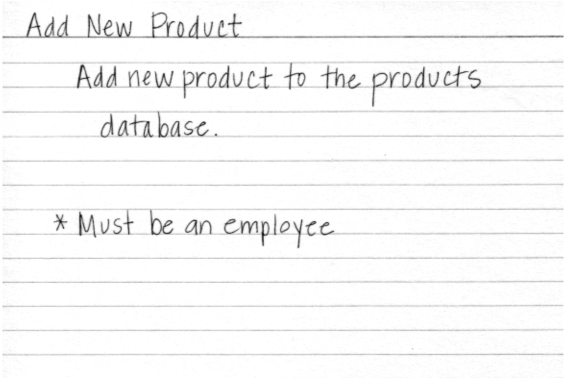
Customer: At this point, I only want to add and edit products.

XP team: Again, that sounds like two stories. Which feature would you like to begin with?

Customer: Let's begin with adding a new product.

XP team: Okay, can anyone add a new product, or do we need to restrict this feature to a particular role?

Customer: Good catch—only employees can add new products. Do you think this captures enough information?



Add New Product

Add new product to the products database.

* Must be an employee

XP team: That will do fine. How about editing products?

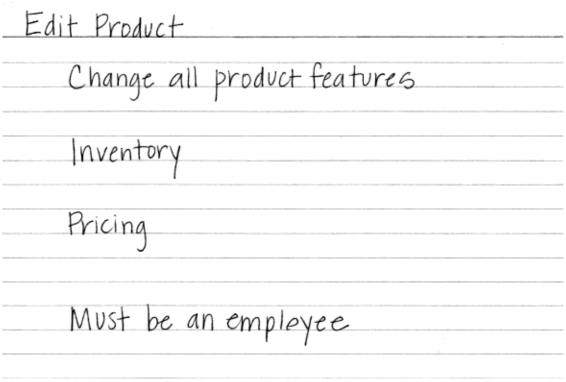
Customer: It is exactly like Add New Product, except that you're changing the attributes of an existing product.

XP team: Do you want to be able to edit all of a product's attributes?

Customer: I really don't know at this point. I do know that I want to be able to change price and inventory, but I'm not sure about the rest. Can I defer until later in the project?

XP team: Yeah, that's no problem. Just make sure you write down those attributes that you are sure about at this point.

Customer: Okay, I think I have it. What about this?



Edit Product

Change all product features

Inventory

Pricing

Must be an employee

XP team: That's great for now. We can further qualify it when we get to the iteration.

Customer: I think that's it. Can you think of anything that I may have left out?

XP team: Let's see. You begin with a login, you browse the catalog displaying lists of products for each category. . .wait, what about selecting a product? I don't think we have that defined.

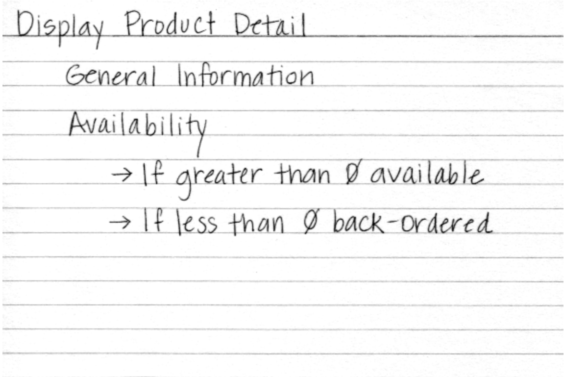
Customer: You're right. We haven't defined that functionality. I guess that would be something as simple as Display Product Detail.

XP team: That sounds good, but what product attributes do you want to display?

Customer: At this point, just the general product information. Can I wait to get descriptive?

XP team: Sure. One thing that we do need to discuss is the inventory. Do we want to display the number of items in stock?

Customer: No, I think we should show the product as available if there is more than one item in stock, or show it as back-ordered if the product has a negative or zero inventory.



Display Product Detail
General Information
Availability
→ If greater than Ø available
→ If less than Ø back-ordered

XP team: Can you think of anything else?

Customer: Not at this point. I think we have a pretty good first pass. What happens if I come up with new functionality?

XP team: That's no big deal. We'll create a new story card and adjust the release, if necessary.

Creating a High-Level Design

The XP team members then talk among themselves and do a very quick high-level design. They first talk about the flow of processing when a user orders a product. Then they walk through the administrative features of the system and the account management features. The result is some high-level diagrams that show screen flow through the system, as illustrated in Figures 11-2, 11-3, and 11-4.

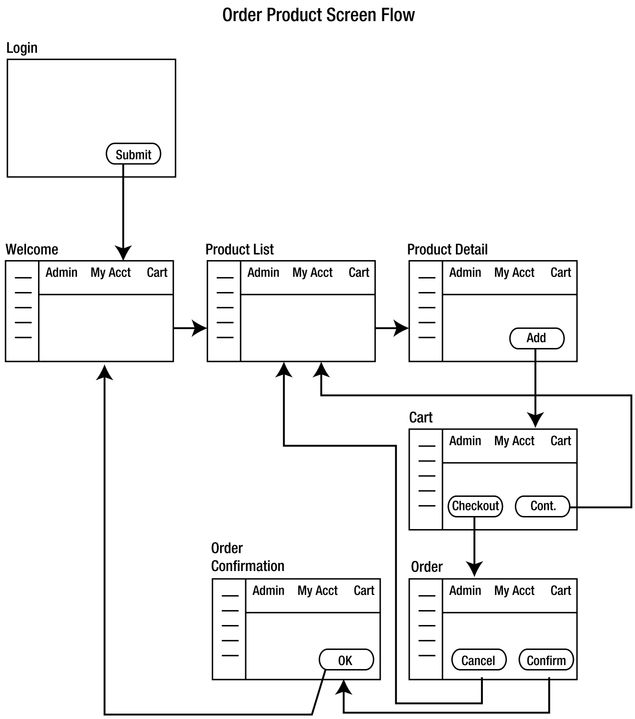


Figure 11-2. *A high-level design of ordering a product*

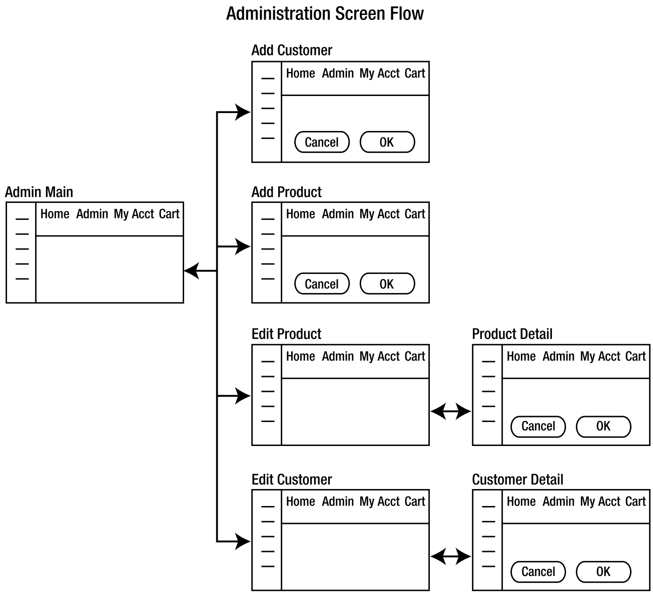


Figure 11-3. *A high-level design of administrative features*

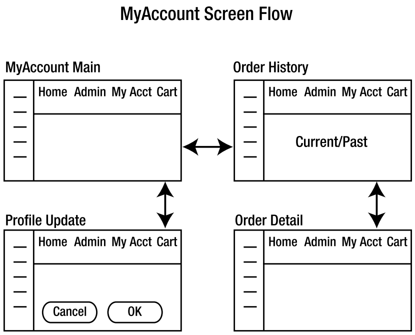


Figure 11-4. *A high-level design of account management features*

Comparing the Stories with the Mission

Now that we have a first pass at our user stories, we need to make sure that our stories match our previously defined mission statement, repeated here:

> *Create a Web presence that allows customers to self-order Northwind products and track the status of their orders all the way through to shipment.*

The conversation may go something like this:

XP team: Do you think that we've sufficiently covered all of the requirements defined by our mission statement?

Customer: Yes, I think we have. We can browse and search the catalog, we can order products, and we can track those products through shipping. I feel like that about covers it.

XP team: Great. Let's move on.

Story Estimating

The next stage in the exploration phase is to begin estimating the level of effort required to complete the user stories. We will review each user story for this purpose. In the process, the testers are going to be asking themselves and the customer if the user story is testable.

If we don't know how to estimate a story, we will spike it. Recall from Chapter 3 that a spike is when the team needs to gain more understanding about the technology or implementation of the story before the story can be estimated.

All estimates are in story points, which are ideal days in length. As a general rule, we recommend that you don't go smaller than a quarter of a story point on any given user story.

Story estimating starts by having the customer read each story again. The developers discuss at a high level what they think they will need to do in order to complete the story. The testers discuss what the acceptance test will be at a high level. If anyone starts diving too deep into the details of a user story, as is often the case, the XP coach will steer the conversation back on track by refocusing the team to the higher-level details. The goal here is to get just enough of an idea as to what each feature (user story) will cost, so the customer can decide if the feature is worth keeping without spending too much time and money arriving at the answer.

Reviewing Each Story

Let's go through each story and estimate its story points.

Login: This user story is very straightforward. The development team and the testers have designed this type of feature many times in the past. The team knows it will need to enhance the existing database a little, and that the login screen will be very simple. Authentication will be the biggest amount of work in this user story. After a brief discussion, we assign an estimate of 1.0 story points.

Add New Customer Account: This user story is also a very common feature. There will be very little database work here, and the entry screen is straightforward, too. We give this story an estimate of 3.0 story points.

Edit Customer Account: This user story is much like the Add New Customer Account user story. Initially, the team talks about leveraging the work from the Add New Customer Account user story, but the XP coach reminds them that it will be the order in which the user stories will be completed is the customer's decision. The estimate should be based on doing the user story all by itself. If it turns out later that the team can leverage work already completed for, say, the Add New Customer Account feature, then the team will finish this Edit Customer Account story earlier than estimated, and they will go back to the customer and ask for more work. Also, there is a difference here, where the user as well as someone with administrative rights can execute this feature. The administrator can edit any customer's data, but the users can edit only their own data. This will add to the complexity of the feature. So, the team talks some more and arrives at an estimate of 3.5 story points.

Add New Product: For this user story, the team members discuss that they estimated the Add New Customer Account user story as 3.0 story points and that this feature is about the same degree of difficulty and effort. Therefore, we give this user story an estimate of 3.0 story points also.

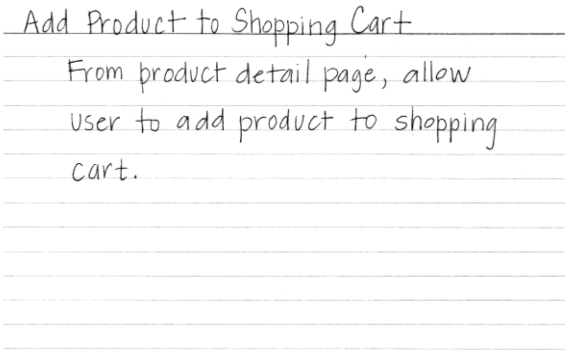
Edit Product: Only users of the application with administrative rights can execute this feature. This will make the processing of this feature simpler than the Edit Customer Account feature. However, a product is more complicated than a customer record. We give this feature an estimate of 3.5 story points.

Search for Product: This user story is considered more difficult and complex than the previous user stories. There is concern that a given search result could return a significant amount of data that will need to be managed correctly so that the application is scalable, as the product data increases over time. The team feels that the level of effort and difficulty is more than twice as hard as the edit features previously discussed. We agree on an estimate of 7.0 story points.

Browse Catalog: This user story also presents some difficulty, as page geometry can get tricky when creating a balanced presentation. The team estimates this feature at 4.0 story points.

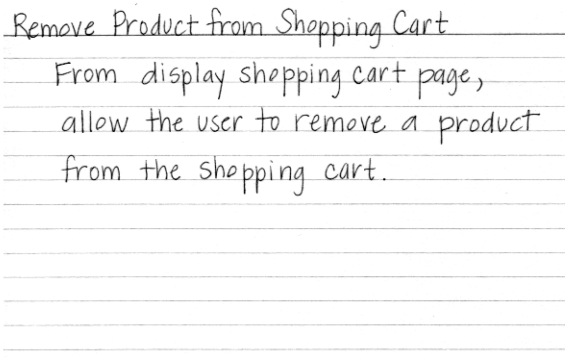
Display Product Detail: This user story is easier than the Browse Catalog user story. The team gives this feature an estimate of 2.0 story points.

When the customer reads the next user story, Shopping Cart, the team members confer and decide that the story is really four features in one. They ask the customer if the user story could be split into four cards: Add Product to Shopping Cart, Remove Product from Shopping Cart, Update Shopping Cart Contents, and Display Shopping Cart Contents. The customer agrees, creates the new user story cards, and tears up the originals so they will not be confused with the existing valid user stories.



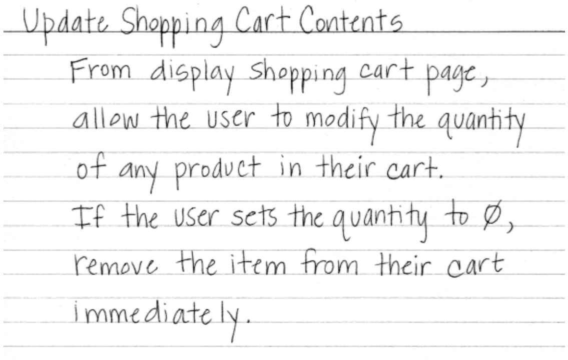
Add Product to Shopping Cart

From product detail page, allow user to add product to shopping cart.



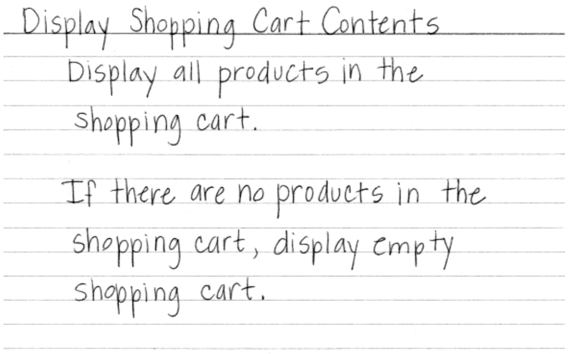
Remove Product from Shopping Cart

From display shopping cart page, allow the user to remove a product from the shopping cart.



Update Shopping Cart Contents

From display shopping cart page, allow the user to modify the quantity of any product in their cart.
If the user sets the quantity to Ø, remove the item from their cart immediately.



Display Shopping Cart Contents

Display all products in the shopping cart.

If there are no products in the shopping cart, display empty shopping cart.

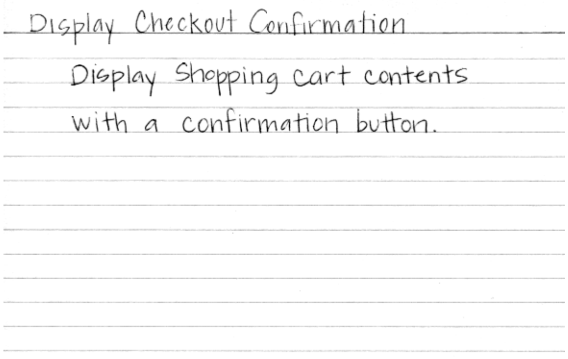
Add Product to Shopping Cart: This user story looks simple, but the processing that needs to happen behind the scenes for updating the cart's contents will require some more database work. Also, the customer overhears the developers' and testers' conversation and realizes that he will need a visual update to indicate to the user that the selected product was successfully added to the cart. For now, this is just added to the user story. The estimate is 2.0 story points.

Remove Product from Shopping Cart: This user story is fairly simple. There isn't much discussion here. The team estimates 1.5 story points.

Update Shopping Cart Contents: This user story gets a little tricky, because if the user enters a quantity of zero, the same action taken for the Remove Product from Shopping Cart needs to happen. The estimate the team gives is 2.5 story points.

Display Shopping Cart Contents: This user story is very simple. The team estimates 1.0 story points.

After the customer rereads the Check Out user story, the developers and testers start to talk and decide that the intent of the user story is really to display to the users their new order and ask the users to confirm their order before processing the order. The customer overhears their conversation and chimes in, agreeing with their conclusion. The customer rewrites the user story with the title Display Checkout Confirmation.



Display Checkout Confirmation

Display Shopping cart contents with a confirmation button.

Display Checkout Confirmation: The team comes to the agreement that the estimate on this new story is 1.0 story point.

Display Order Confirmation: This user story sounds like a lot of processing. The team members have not done a feature like this before, but they know someone else outside the team who has. They give a quick call to that person and start to get a feel for how big this feature really is. With the knowledge the team gained by talking with the person outside the group about this feature, the team gives an estimate of 5.5 story points.

Display Order Status: This user story looks like there might be a bit of back-end processing and verification going on. So, the team estimates this feature at 2.0 story points.

Display Order History: This user story appears to be about half the degree of difficulty and effort as the Display Order Status story. Therefore, the team uses the estimate of 1.0 story points.

Getting the Big Picture

The user stories have all been estimated, so that the customer now understands what the cost of implementing each feature will be. These are intentionally high-level estimates, to allow everyone to start to understand the size of this project and determine if the project is worth the investment. We are making this determination as soon and as quickly as possible to minimize the upfront costs of the project as much as possible.

Table 11-1 summarizes all of our stories and their associated estimates. Note that we are listing the user stories in a table simply for clarity in the book. In reality, you will write the estimates on the cards and lay the user stories out on a table or wall to get the “big picture” view.

Table 11-1. *Estimated User Stories*

User Story	Estimated Story Points
Login	1.0
Add New Customer Account	3.0
Edit Customer Account	3.5
Add New Product	3.0
Edit Product	3.5
Search for Product	7.0
Browse Catalog	4.0
Display Product Detail	2.0
Add Product to Shopping Cart	2.0
Remove Product from Shopping Cart	1.5
Update Shopping Cart	2.5
Display Shopping Cart Contents	1.0
Display Checkout Confirmation	1.0
Display Order Confirmation	5.5
Display Order Status	2.0
Display Order History	1.0
Story Point Total	43.50

Declared Velocity

Now the planning game phase begins. The developers and testers talk to the customer about how often the customer would like to see progress during development. They tell the customer that they can deliver new functionality in as little time as every week and as long as every three weeks, but every two weeks is ideal. The customer agrees that delivery of new features every two weeks is good for him.

Next, the customer asks when all the features described can be delivered. Because this is a new development team, without any XP team history, the team's tracker does a quick *one-time* calculation. There are four developers on the team and it was just decided that the team will deliver in two-week iterations. The calculation shows that this team will deliver all the requested features in four iterations or two months, as follows:

1. Calculate the velocity of a single iteration:

   (4/3 × 10) (Truncate) = 13

2. Divide the total number of user story points by the number of story points in a single iteration, and then round up, because the team does full iterations:

   (44.5/13) (Round Up) = 4

The customer states that the projects must be delivered in four weeks, or two iterations, because of market demands and pressure from the executive management groups. The customer asks the developers and testers to please work harder and get all the work done in four weeks. The developers and testers tell the customer that they will, of course, work as hard as they can. If they finish early, they will come back and ask for more work, but if they overcommit now, the team will be setting themselves up for failure from the start.

The developers and testers ask the customer to please select a subset of the user stories that will not exceed the number of story points they can complete in two iterations, which, in this case, is 26 story points.

Story Selection

At this point, we have our user stories, we know our velocity, and we know how long our release is going to be. We get down to business and select the features that will be in our first release.

According to the calculations from the previous section, our velocity provides us with 26 ideal days to accomplish our first release, but we have 43.5 ideal days of work to do, so something is going to be pushed to the next release. It is now the job of the customer to prioritize the user stories and select the ones that provide the most business value for this release.

Prioritizing the Stories

The first round of story prioritization might begin like this:

Coach: The first thing you need to do is prioritize your stories.

Customer: Okay, give me a few minutes. I'm going to rearrange my story cards into the priority that will provide me the most value. . . . I believe I have it. The main thing is that my users must have the ability to browse and submit orders.

The customer comes up with the priorities shown in Table 11-2.

Table 11-2. *Prioritized User Stories*

Story	Story Points
Login	1.0
Browse Catalog	4.0
Display Product Detail	2.0
Search for Product	7.0
Add Product to Shopping Cart	2.0
Remove Product from Shopping Cart	1.5
Update Shopping Cart	2.5
Display Shopping Cart Contents	1.0
Display Checkout Confirmation	1.0
Display Order Confirmation	5.5
Display Order Status	2.0
Display Order History	1.0
Add New Customer Account	3.0
Edit Customer Account	3.5
Add New Product	3.0
Edit Product	3.5
Story Point Total	43.50

Selecting a Subset of Stories

Next, the customer selects the subset of stories that the team should complete in the first release.

Coach: Okay, now select a subset of your defined user stories that doesn't total more than 26 story points.

Customer: Okay, I have removed as many user stories as possible, but I am still over my point total by 4.5 points.

Table 11-3 shows the customer's dilemma.

Table 11-3. *First Subset Cut*

Story	Story Points
Login	1.0
Browse Catalog	4.0
Display Product Detail	2.0
Search for Product	7.0
Add Product to Shopping Cart	2.0
Display Shopping Cart Contents	1.0
Remove Product from Shopping Cart	1.5
Update Shopping Cart	2.5

continued

Table 11-3. *Continued*

Story	Story Points
Display Checkout Confirmation	1.0
Display Order Confirmation	5.5
Display Order Status	2.0
Display Order History	1.0
Story Point Total	30.50

Refining the Subset Selection

Now we need to help the customer arrive at the best solution.

Customer: What can I do?

Coach: You have a few options here. You can juggle your priorities so that you have exactly 26 points. You can just remove a user story that totals 4.5 points or more. Or you can try to simplify one of your existing user stories.

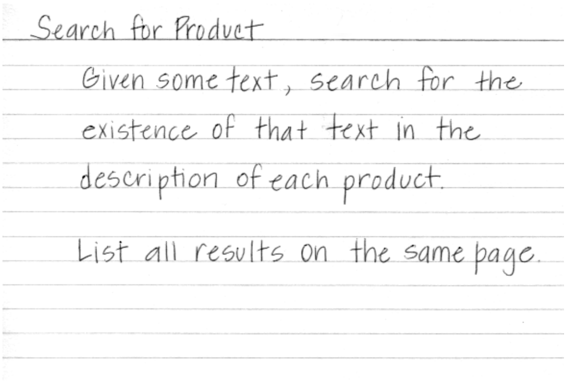
Customer: I really don't want to change my priorities, and I don't want to remove any items. What do you suggest?

Coach: Well, I think I would try to pare down the Search for Product story. It is your largest user story. We can probably keep the basic functionality, while reducing the level of effort. Let's ask the developers what they think. Developers?

Developers: The largest effort involved in this story is handling multiple pages of search results. If we can display all of the search results on a single page and restrict the search to the Product Name field, then I think we can reduce this story to 1.5 ideal days.

Coach: How does that sound?

Customer: Great. I can work with this. I'll rewrite the story card.



Search for Product

Given some text, search for the existence of that text in the description of each product.

List all results on the same page.

Coach: Okay, we are at exactly 25 ideal days. This leaves us 1 story point short. Is there any outstanding user story that equals exactly 1 point?

Customer: No, I can't find anything. I don't even see a story that I can simplify. What happens to my extra point?

Coach: If we finish either iteration early, we'll look at adding some additional work.

Table 11-4 shows the complete list of user stories selected for this release.

Table 11-4. *User Stories for the First Release*

Story	Story Points
Login	1.0
Browse Catalog	4.0
Display Product Detail	2.0
Add Product to Shopping Cart	2.0
Remove Product from Shopping Cart	1.5
Update Shopping Cart	2.5
Display Shopping Cart Contents	1.0
Display Checkout Confirmation	1.0
Display Order Confirmation	5.5
Display Order Status	2.0
Search for Product	1.5
Display Order History	1.0
Story Point Total	25

Coach's Journal

We started our project. We had our customer, developers, and testers together in one room to go through the release plan. They were all a little leery about being together. After introductions and an overview of the project's mission and objective, they all started to relax a little.

When we started story writing, the customer looked really nervous. Fortunately, the customer seemed to get the hang of story writing quickly, and the developers and testers were good about giving suggestions. Sometimes, the developers got down too much into the weeds, and I had to pull them out of the details. They reacted well though and were good about not getting wrapped up in too many details before giving high-level estimates.

■ **Note** In reality, everyone on the team should keep a journal and write in it every day. In order to keep this book as simple as possible, we will journal from only the development coach's perspective, and we will include only one journal entry per chapter.

Summary

In this chapter, we began the journey that will take us through the rest of the book. We started with an introduction to the business problem. We explored the business problem and created user stories along the way. Then we assigned a cost to each of the user stories.

We then started the planning game portion of the release plan. The team's tracker declared the amount of work the team could sign up for, without overcommitting the team's time. Harnessed with this information, the customer was able to prioritize the user stories based on business need and cost, and then select the subset of user stories that can be completed by the release date. This subset of stories makes up the release that we will develop over the rest of the book.

All of these activities were accomplished as a team—customer, developer, tester, and so forth. This creates a highly collaborative environment that allows the team to work smarter and faster. Release planning will normally occur over the course of three to five days instead of the traditional weeks and months needed for analysis.