

# BECOMING AGILE

in an imperfect world



GREG SMITH  
AHMED SIDKY  
FOREWORD BY  
MARY POPPENDIECK



***Becoming Agile***  
*... in an imperfect world*

by Greg Smith  
and Ahmed Sidky

Chapter 13

Copyright 2009 Manning Publications

# *brief contents*

---

<b>PART 1</b>	<b>AGILE FUNDAMENTALS AND A SUPPORTING CASE STUDY.....</b>	<b>1</b>
	1 ■ Moving to agile	3
	2 ■ The story of Acme Media	17
<b>PART 2</b>	<b>GETTING STARTED.....</b>	<b>25</b>
	3 ■ Are you ready for agile?	27
	4 ■ The fitness test: all about readiness assessments	43
	5 ■ The importance of obtaining executive support	58
	6 ■ Improving buy-in by creating a core team	66
	7 ■ The mindset of an agile leader	73
	8 ■ Injecting agility into your current process	87
	9 ■ Selecting a pilot project	105
<b>PART 3</b>	<b>KICKING OFF.....</b>	<b>113</b>
	10 ■ Feasibility: is this project viable?	115
	11 ■ Aligning the pilot team with the project	136

<b>PART 4</b>	<b>POPULATING THE PRODUCT BACKLOG.....</b>	<b>151</b>
	12 ■ Feature cards: a tool for “just enough” planning	153
	13 ■ Prioritizing the backlog	170
	14 ■ Estimating at the right level with the right people	183
<b>PART 5</b>	<b>ENOUGH INFORMATION FOR SCHEDULING.....</b>	<b>193</b>
	15 ■ Release planning: envisioning the overall schedule	195
	16 ■ Iteration planning: the nitty-gritty details	204
<b>PART 6</b>	<b>BUILDING THE PRODUCT .....</b>	<b>221</b>
	17 ■ Start your engines: iteration 0	223
	18 ■ Delivering working software	230
	19 ■ Testing: did you do it right?	244
<b>PART 7</b>	<b>EMBRACING CHANGE.....</b>	<b>253</b>
	20 ■ Adapting: reacting positively to change	255
	21 ■ Delivery: bringing it all together	277
	22 ■ The retrospective: working together to improve	297
<b>PART 8</b>	<b>MOVING FORWARD.....</b>	<b>311</b>
	23 ■ Extending the new process across your company	313

# *Prioritizing the backlog*

---



Have you ever run out of time on a project and started skimping on testing, training, or usability? Have you ever delivered a project and then been amazed that the customer only used about 20 percent of the functionality? These common issues illustrate the need for prioritizing features within a project.

When you prioritize features, you ensure that you deliver the most valuable functionality to your customer first. You do this by iteratively building feature sets and deploying these features after each iteration if needed. For example, you may find that you need three iterations of work to complete your project. In this instance, you put the critical, minimum functionality needed in iteration 1, followed by high-priority features in iteration 2, followed by medium-priority features

in phase 3. Each iteration concludes with usable features that could be deployed to your production environment if needed. The power in this approach is you can still deliver critical parts of your project if issues are encountered along the way. You won't need to compromise on the quality of the features you deliver, and the customer can still receive a usable system.

In this chapter, we'll discuss the guidelines for prioritizing features, and you'll see these guidelines in action as Acme Media prioritizes its project/product backlog for the Auctionator project. At the conclusion of this exercise, Acme Media will have a prioritized backlog that is ready for initial estimation.

### **13.1 The art of prioritizing, sequencing, and grouping features**

After you complete the feature-card exercise, you need to determine the logical build sequence. There are dozens of ways to do this; we'll walk you through a process that has worked well for many project teams in various industries.

However your project team chooses to do this process, follow the way you do it consistently so your team can focus on building the application. You want the team to get familiar with a sequencing process so that it becomes intuitive.

Here are the guidelines we suggest for prioritizing, sequencing, and grouping features:

- Determine what features are most important to the user/customer.
- Determine which features go together to provide value to the customer. If possible, avoid splitting features across groupings when both features are needed to provide value to the user.
- Think about features that are high value and high risk. It's usually good to get them up front in the sequence so you can make the call on removing them from the project or have the luxury of being able to work on them in every iteration.
- Interfaces always present risk. As a rule of thumb, feature cards that involve an interface should have technical uncertainty marked as *high*.
- Third parties (vendors/partners) also represent high risk, especially if you've never worked with them before. You want features related to third parties early in the sequence also.

After you have the guidelines for prioritizing, you need to determine who will take part in the prioritization exercise. In most agile environments, the customer or product owner prioritizes the backlog. We agree with this practice for initial prioritization, which is based on anticipated business value. But after the customer has prioritized the features, you should engage the entire project team in an exercise of reviewing the priorities based on additional attributes of the feature cards, such as frequency of use, technical/requirements uncertainty, and dependencies.

Our suggestion is to tape all the feature cards to a wall, in the order of business value, and then have the team move the cards around. The highest-priority items are on the left, and the lesser-priority items are on the right. The team looks at the



**Table 13.1 Acme Media has a product backlog after completing the feature-card exercise. The work hasn't been prioritized yet, but the team has the information they need. Note that you should assign unique IDs to features to prevent confusion when you have features with similar names. (continued)**

Customer value	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Critical	4	Place an item up for bid	Low	Low	High	5	
Critical	5	Register on the site	Low	Low	High		Ties into existing registration functionality
High	6	Contact the seller	Low	Low	Medium	4, 5	
Low	7	Customize my view	Low	Low	Low	5	
Low	8	Use an auction browser toolbar	Low	High	Low	5	No experience with browser toolbars
Medium	9	Receive help online	Low	Low	Medium		
Critical	10	Bid on an item	Low	Low	High	4, 5	
Medium	11	Record seller feedback	Low	Low	Medium	5	
Medium	12	View seller information	Low	Low	Medium	11	
Critical	13	Search by category	Low	Low	High		
Medium	14	Perform advanced search	Low	Low	Low	4	
Low	15	Retract a bid	Low	Low	Low	5	
Medium	16	Purchase an item immediately	Low	Low	High	5	
Critical	17	Auction engine	Medium	Low	High	4, 10	Processes to support the auctions

Note that we've aggregated the feature information into a table to make it easier for you to follow the process. At Acme Media, the team uses their hard-copy cards and the wall for the prioritization exercise.

### 13.2.1 Prioritizing by value

Acme Media's first step is to make sure the business priorities are unchanged after the feature-card exercise. The customer may want to reassign the priorities now that you have more information about the features.

The Acme Media project team asks their customer, Jay, if he still agrees with his initial value assessments. Jay notes that the *purchase an item immediately* feature is predicted to have high usage. Looking at it now, he'd label it as high or critical business value. Jay decides to mark the feature *high*.

After making the change that Jay requests, the team sorts the cards by customer/business value. You can see the results from the initial sorting in table 13.2.

**Table 13.2** Feature cards sorted by customer value. You sort by customer value first because delivering valuable software “early and often” is your main objective.

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Critical	4	Place an item up for bid	Low	Low	High	5	
Critical	10	Bid on an item	Low	Low	High	4, 5	
Critical	5	Register on the site	Low	Low	High		Ties into existing registration functionality
Critical	13	Search by category	Low	Low	High		
Critical	17	Auction engine	Medium	Medium	High	4, 10	Processes to support the auctions
High	16	Purchase an item immediately	Low	Low	High	5	
High	1	Flag problem postings	Low	Low	Low	4	
High	6	Contact the seller	Low	Low	Medium	4, 5	
Medium	2	Create alerts for item type	Low	High	Low	4	Feature requires an interface
Medium	9	Receive help online	Low	Low	Medium		
Medium	11	Record seller feedback	Low	Low	Medium	5	

**Table 13.2 Feature cards sorted by customer value. You sort by customer value first because delivering valuable software “early and often” is your main objective. (continued)**

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Medium	12	View seller information	Low	Low	Medium	11	
Medium	14	Perform advanced search	Low	Low	Low	4	
Low	3	Email a friend	Low	Low	Low	4	
Low	7	Customize my view	Low	Low	Low	5	
Low	8	Use an auction browser toolbar	Low	High	Low	5	No experience with browser toolbars
Low	15	Retract a bid	Low	Low	Low	5	

This first sort of the feature cards lets the team understand which features are critical, high, medium, and low. Note that another way of viewing the critical features is to say they’re the minimum features needed to deliver a functional application. Although you aren’t creating a release plan yet, the *critical* label gives you early insight into what the first iteration will probably contain.

The next sort relates to dependencies. Which features have a high dependency placed on them? What features tie to each other?

The Acme Media team looks at the Dependency column and sees that eight features are dependent on feature 5, *the ability to register on the site*. As a matter of fact, the only critical feature that doesn’t require user registration is *search by category*. Based on this, the team moves *the ability to register on the site* to the far left, indicating it’s the first feature in the sequence.

### 13.2.2 Evaluating risk

The Acme Team follows the dependency evaluation by reviewing feature risk. The team looks for high-risk features, knowing that they must be removed from the project or moved to the top of the sort.

You have three options for dealing with high-risk features. Here are our suggestions:

- 1 *High risk and critical/high business value*—Move the features to the top of your sort, and pursue them early in the project. By doing this, you’ll be able to use the entire length of the project to work out issues related to the risk.
- 2 *High risk and medium business value*—You have to make the call on a case-by-case basis. You can work on the feature early, in the middle of the project, or at the end of the project. We suggest that you review the value again with the customer

and see which way the feature is leaning. If the customer leans toward high value, pursue option 1. If the customer leans toward low business value, pursue option 3.

- 3 *High risk and low business value*—Move the feature to the bottom of your sort, or consider removing it from the project.

For us, risk is tied to uncertainty. Which features do you have doubts about? Are the requirements still vague for any features? Are you questioning what technology to use for any features?

If you do a quick review of Acme Media’s list, you can see a few features that meet this uncertainty criterion. Feature 17, the *auction engine*, has medium requirements uncertainty and medium technical uncertainty. This feature is also considered critical.

Ideally, Acme would like to move this feature closer to the top, but it’s dependent on other features. The *ability to place an item up for bid* (4) and the *ability to bid on an item* (10) need to be understood at some level before Acme can build the auction engine. Based on this, they move this feature above search but below features 4, 5, and 10.

Another feature with a risk implication is *ability to use an auction browser toolbar* (8). This feature has high technical risk because the development team has never tried to embed functionality into a browser. This feature is also marked as a low priority to the customer. High-priority / high-risk features go to the top of the sequence list. Low-priority / high-risk features go to the end of the sequence. Acme makes the auction browser toolbar the last item in the sequence list.

The last item with a level of risk is *the ability to create alerts for item type* (2). This feature is labeled as high risk because it requires an interface to a third party. Acme has used this vendor in the past and is familiar with the vendor’s API. The risk is probably lower than indicated, so the team leaves this feature where it is in the sequence.

Let’s review Acme’s sequence after these changes; see table 13.3.

**Table 13.3** Feature cards sorted by customer value and risk. You sort high-value/high-risk items to the top to ensure time for resolving the risk. You sort high-risk/low-value items to the bottom.

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Critical	5	Register on the site	Low	Low	High		Ties into existing registration functionality
Critical	4	Place an item up for bid	Low	Low	High	5	
Critical	10	Bid on an item	Low	Low	High	4, 5	

**Table 13.3 Feature cards sorted by customer value and risk. You sort high-value/high-risk items to the top to ensure time for resolving the risk. You sort high-risk/low-value items to the bottom. (continued)**

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments	
Critical	17	Auction engine	Medium	Medium	High	4, 10	Processes to support the auctions	
Critical	13	Search by category	Low	Low	High			
High	16	Purchase an item immediately	Low	Low	High	5		
High	1	Flag problem postings	Low	Low	Low	4		
High	6	Contact the seller	Low	Low	Medium	4, 5		
Medium	2	Create alerts for item type	Low	High	Low	4		Feature requires an interface
Medium	9	Receive help online	Low	Low	Medium			
Medium	11	Record seller feedback	Low	Low	Medium	5		
Medium	12	View seller information	Low	Low	Medium	11		
Medium	14	Perform advanced search	Low	Low	Low	4		
Low	3	Email a friend	Low	Low	Low	4		
Low	7	Customize my view	Low	Low	Low	5		
Low	15	Retract a bid	Low	Low	Low	5		
Low	8	Use an auction browser toolbar	Low	High	Low	5	No experience with browser toolbars	

Now Acme’s team has a good sort of their features. Critical features are first in the list, where they belong. Risky features have been sorted accordingly and either moved up front to buy the team time to work out the risk or moved to the end because they have low customer value.

### 13.2.3 Grouping related features

The last step in Acme Media’s sorting exercise is grouping features. Acme needs to group features that must be used together to provide value to the user/customer.

For example, the customer won’t care if Acme delivers the *ability to bid on an item* (10) if no one has the *ability to put an item up for bid* (4), because there won’t be any items to bid on. These two features need each other to provide value to the customer. Acme places them together and calls them Group A.

If you review the features, another logical grouping stands out. Users can’t *view seller information* (12) if they don’t have the *ability to record seller feedback* (11). These two features needed to be delivered together to provide value to the customer (Group B).

Let’s take a final look at the feature list after Acme establishes the groupings; see table 13.4.

**Table 13.4 Feature cards sorted by customer value and risk, with logical groupings. Groupings show you features that must be delivered together to provide value to the customer.**

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Critical	5	Register on the site	Low	Low	High		Tie into existing registration functionality
Critical	4	Place an item up for bid	Low	Low	High	5	Group A
Critical	10	Bid on an item	Low	Low	High	4, 5	Group A
Critical	17	Auction engine	Medium	Medium	High	4, 10	Processes to support the auctions
Critical	13	Search by category	Low	Low	High		
High	16	Purchase an item immediately	Low	Low	High	5	
High	1	Flag problem postings	Low	Low	Low	4	
High	6	Contact the seller	Low	Low	Medium	4, 5	
Medium	2	Create alerts for item type	Low	High	Low	4	
Medium	9	Receive help online	Low	Low	Medium		
Medium	11	Record seller feedback	Low	Low	Medium	5	Group B

**Table 13.4 Feature cards sorted by customer value and risk, with logical groupings. Groupings show you features that must be delivered together to provide value to the customer. (continued)**

Customer priority	ID	Feature name (ability to)	Requirements uncertainty	Technical uncertainty	Usage	Dependency	Comments
Medium	12	View seller information	Low	Low	Medium	11	Group B
Medium	14	Perform advanced search	Low	Low	Low	4	
Low	3	Email a friend	Low	Low	Low	4	
Low	7	Customize my view	Low	Low	Low	5	
Low	15	Retract a bid	Low	Low	Low	5	
Low	8	Use an auction browser toolbar	Low	High	Low	5	No experience with browser toolbars

Now that the team has determined their priorities, sequence, and groupings, it's time to estimate the features. This process is covered in chapter 14.

We suggest that you hold off on estimating features until you've completed the sequencing exercise. The reason is you may identify several low-value items that you may want to remove from the project. If you remove them, you don't need to waste time estimating them.

**You can use estimates to assist in prioritizing the backlog**

As noted in this chapter, we suggest performing initial estimates after you've determined the priority of each feature. Our suggestion is based on the time needed to estimate each feature. You don't want to waste time estimating a feature that you may never pursue.

In the last few years, however, we have crossed paths with teams who do a quick initial estimate and use it as another factor in the sequencing/prioritization process we've outlined in this chapter. We don't have an issue with this approach, as long as you time-box the estimating window. It's easy for a team to want to do a first pass at designing features before providing estimates. This can be a time-consuming process and may delay the start of developing features that are known to be critical to the project.

Working with the customer, you can review the sequenced features and see if any low-priority features should be removed. You especially want to consider features that are low value and high risk. In the case study, one feature stands out: the *auction browser toolbar* (8). It's low value and high risk. The Acme Media team discusses its value and risk with Jay, the customer, and he agrees to remove the feature from the project. Jay

understands how the feature could be technically complex and distract the team from the most important features.

### 13.3 Other ways to prioritize features

As we mentioned at the start of this chapter, there are many ways to prioritize features. Section 13.2 demonstrates a common way to perform prioritization; let's contrast it to another approach.

We have a friend, Tim, who once worked for a small startup company that was fighting for survival. The company provided commercial software to other companies; similar to Acme Media, the company delivered to a product market, not a specific customer.

Tim's team performed the feature-card exercise but used a slightly different approach for determining feature value (see figure 13.2). Tim's company looked at three factors when determining feature priority:

- How close is this feature to the target market? This is the most important ranking attribute; multiply the score of this value by 4.
- How much effort will this feature take to complete? Multiply this score by 3.
- How much organizational impact will this feature have? In this instance, *organizational impact* means the feature will make the company more attractive to potential investors.

Notice that Tim's team didn't score the items 1 to 10. Every item had to be scored as 9, 3, or 1. This scoring method was used to push the team to start thinking about priorities immediately and to avoid having all items reach a similar score.

When Tim's team concluded this exercise, they didn't consider the sorting process complete. They used the output of the Excel spreadsheet as a starting point for dialogue on how they would prioritize the items. The scores influenced their decisions, but ultimately the team made the call about priorities.

Customer Facing Features	Ratings			Total
	Market (x4)	Effort (x3)	Org Impact (x1)	
External				
User-purchased gift certificates	9	9	9	72
Buy-X get Y free promotion	9	9	9	72
Save-your-shoppingcart	9	3	9	54
Promotion Mgmt.	9	3	9	54
Menu of 3-4 different page flows (express, etc)	9	1	9	48
OrderHistory - Change ship-to address prior to export	3	9	9	48
Single-purchase coupons	3	9	9	48
Enhanced Billing UI	3	9	9	48
Affinity points	9	3	1	46
Dynamic personalized upsells/xsells	9	1	3	42
VIPC Lite	1	9	9	40
OrderHistory - Self-service returns	3	3	9	30
Buy-now, ship-later	3	3	3	24
Gift Cards & personalized messages	3	3	3	24
Gift Wrap	3	3	3	24
Wish Lists	3	3	3	24
WAP/WML support	3	1	9	24
HTML Emails	1	3	9	22
Auctions	3	1	3	18
OrderHistory - show "real" shipments	1	3	3	16
Configurable Multiple Ship To	1	3	2	15

Figure 13.2 An Excel spreadsheet is used to weight and sort features for a release.

### 13.3.1 What about technical features?

You may have noticed that most of the features/user stories in the Auctionator are customer facing. We've done this to make it easier to follow the case study, but we understand that every project/release you perform includes some level of technical work, and this work must be prioritized with the customer-facing features.

Returning to Tim's small company, figure 13.3 shows how his team prioritized their technical/refactoring features.

Refactoring Features	Ratings			Total
	Benefit (x4)	Effort (x3)	Org Impact (x1)	
Internal				
Email service	9	9	1	64
Table shipping algorithm service	9	3	9	54
HTML/UI friendliness cleanup and enhancements	9	3	9	54
Modularize promotion arbitration process	3	9	9	48
Close existing bugs before adding new features	3	9	9	48
Refactor abstract servlet hierarchy	6	6	3	45
Order-placement and payment-processing service	3	3	9	30
Distributed webcart	3	3	9	30
Multi-site support	3	3	9	30
Allow cart operation even if tax and fraud aren't available	3	3	9	30
Clean up the Validation Infrastructure	3	3	9	30
Simplify/refactor rendering components	3	3	9	30
Rewrite logging functionality	3	3	9	30
Shipping service table algorithm > 150 lbs	3	3	9	30
Completion of FedEx table support	3	3	9	30
International shipping	3	3	9	30
Change taxing to accept multiple items in one network	3	3	3	24
Site/Merchant/Supplier/Shipping configuration gui	3	1	9	24
Shipping calculation service	3	3	1	22

**Figure 13.3** Every project will have features that are related to architecture and refactoring. This work is prioritized with the customer features, and your release will be a mixture of both types of features.

In this example of architectural features, the team used similar attributes to rank the backlog. The main difference was removing the *Market* column and replacing it with a column that measures how valuable the feature is to the platform.

Your technical features will be stored in the same backlog as the customer-facing backlog; but similar to Tim's team, you may use different attributes to rank the features.

Technical features frequently correlate to refactoring work and consolidating redundant programs. In this example, Tim's company had four product groups, and each had created its own email engine. The team realized how much work was being wasted in maintaining four sets of similar code, so they decided to work together to create one email service that they could all use.

## 13.4 Key points

The key points from this chapter are as follows:

- Prioritizing features helps you deliver value to your customer sooner.
- Prioritizing features lets you stop a project before it's complete and still deliver the critical features.

- Customer value is the main attribute for determining prioritization, but you use other factors such as risk, frequency of use, and dependencies to create your final prioritized backlog.
- The customer or product owner can provide the business priorities, and then the team and the customer should work together to complete the sorting process. This step aligns the team on the priorities and contributes to team buy-in.
- Some features need to be delivered together to provide value to the customer. When these features are identified, you group them together, and they have equal priority.
- You can customize the prioritization at your company to meet your unique needs. You should start with customer value and then consider other areas that are of value in your environment. These areas can include market share, usability, feature expense in time and or money, investor value, and innovation.
- Your backlog will consist of features that are customer facing and features that may be considered refactoring. You'll review both types of features when prioritizing your product backlog.

### **13.5** *Looking forward*

In this chapter, we explained how to prioritize features based on business value and technical risk. In chapter 14, we'll show you how to estimate the prioritized work so that you can assign features to iterations.

# BECOMING AGILE ... in an imperfect world

Greg Smith • Ahmed Sidky

Foreword by Mary Poppendieck

People know about the benefits of agility but many have been reluctant to make the transition themselves. This book will help overcome the barriers that hold them back. Whether you're a small shop or part of a large corporation, you can benefit by learning the practices it recommends. They'll make your dev processes faster, more flexible, and more cost effective.

**Becoming Agile** addresses the real issues—including the needs of executives, managers, and the development team during the transition. It starts by helping you gauge how agile you already are. Then, it shows you how to create a process that supports the realities of your environment so you can transition gradually to an agile way of working.

## What's Inside

- Obtain team buy-in for the move to agile
- Assess what practices you can pursue with low risk
- Reach the right level of agility for your organization
- Create an agile process within your specific constraints

**Greg Smith** is a Senior Project Manager, ScrumMaster, and an Agile college instructor. Greg has helped teams become agile at the start-up and enterprise level. **Dr. Ahmed Sidky** is an experienced agile coach, helping agile teams around the world. Called "Dr. Agile" for the creation of his readiness assessment tool (doctoragile.com) and the Sidky Agile Measurement Index.

For online access to the authors, code samples, and a free ebook for owners of this book, go to [www.manning.com/BecomingAgile](http://www.manning.com/BecomingAgile)

"A thoughtful, practical set of steps ... to become agile."

—From the Foreword by  
Mary Poppendieck

"Makes agility rhyme with reality."

—Valentin Crettaz  
Web Architect, Consulthys

"This book is a *necessity* for teams looking to become agile..."

—Randy Miller, Software Project  
Recovery Expert, Microsoft

"The first book I can give to anyone on an agile team, regardless of their role."

—Paul King, Director, ASERT

"A 5-star guide"

—Robert Dempsey, CEO,  
Atlantic Dominion Solutions

