Improving customer service contact root-cause analysis

By

Grant Stephen Elliott

B.S. in Mechanical Engineering, United States Naval Academy, 2002

Submitted to the MIT Sloan School of Management and the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration

AND

Master of Science in Mechanical Engineering

In conjunction with the Leaders for Manufacturing Program at the

Massachusetts Institute of Technology

June 2009

© 2009 Massachusetts Institute of Technology. All rights reserved.

Signature of Author

Department of Mechanical Engineering &
MIT Sloan School of Management
May 9, 2009

Certified by
Sanjay Sarma, Thesis Supervisor
Associate Professor of Mechanical Engineering

Certified by
Stephen Graves, Thesis Supervisor
Abraham Siegel Professor of Management, MIT Sloan School of Management

Accepted by
David Hardt, Professor of Mechanical Engineering
Chairman, Committee on Graduate Students

Accepted by
Debbie Berechman
Executive Director of MBA Program, MIT Sloan School of Management
This page has been intentionally left blank
Improving customer service contact root-cause analysis

By
Grant Stephen Elliott

Submitted to the Sloan School of Management and the
Department of Mechanical Engineering on May 9, 2009 in Partial Fulfillment of the
Requirements for the Degrees of Master of Business Administration and
Master of Science in Mechanical Engineering

ABSTRACT

When a customer calls or e-mails customer service, a customer service agent will diagnose the
issue, render a solution, and then wrap-up the call or e-mail. For many customer service
departments, this wrap-up process requires the agent to classify the reason the customer
contacted customer service. Typically, this classification is done by assigning a code that
describes the reason for a contact. Additionally, if a contact requires a concession, the agent will
classify the reason the customer requires a concession, and select an appropriate code.

These codes are used by the various business teams within the company to identify and correct
failures in their processes. Therefore, these codes should drive down to the root cause for a
contact or concession to allow for efficient correction. Possessing codes that do not clearly
identify the root cause for a contact are of little or no use for the company. Additionally, the
codes must be developed in such a way that they can be accurately chosen by either the agent or
the customer. Having agents select the wrong code not only obscures the true cause for a
contact, but also creates additional work due to the process involved in determining the correct
code.

This thesis looks at the challenges inherent in developing a list of codes that both provides clear
insight into the root cause for customer contacts, and can be accurately selected by the customer
service agent or the customer.

Thesis Supervisor: Sanjay Sarma
Title: Associate Professor of Mechanical Engineering

Thesis Supervisor: Stephen Graves
Title: Abraham Siegel Professor of Management, MIT Sloan School of Management
This page has been intentionally left blank
Acknowledgements

The members of the operational excellence team at Amazon.com: Luis Carbajo, Bryant Condie, Ana Tischler, Frank Haude, and Al Massa provided me with much guidance and support throughout my six months in Seattle.

My thesis advisors, Stephen Graves and Sanjay Sarma, gave me timely and thoughtful feedback during the internship, as well as during the writing of this thesis. Their outside perspective and expertise enhanced this learning experience.

The members of the Leaders for Manufacturing team, especially my fellow classmates, have assisted me in maintaining my sanity through this once-in-a-lifetime opportunity at one of the world’s premier academic institutions.
This page has been intentionally left blank
Biographical Note

Hailing from San Diego, CA, Grant launched his East Coast tour in Annapolis, MD at the United States Naval Academy, where he graduated with distinction in 2002, earning a BS in Mechanical Engineering. Commissioned as a naval officer, he served onboard the USS HARTFORD, a nuclear submarine stationed in Groton, CT. While stationed onboard, he conducted one six-month deployment to the Caribbean and Eastern Pacific, carrying out counterdrug and surveillance operations.

After leaving the Navy in 2007, Grant was somehow accepted into the Leaders for Manufacturing program, a joint-degree program that is a collaboration between the MIT Sloan School of Management and the MIT School of Engineering, where he is currently working on his MBA and MS in Mechanical Engineering.

Grant has accepted an offer at Gotham Consulting Partners, an operational consulting firm, in New York City, and will be starting after he graduates in June 2009.
# Table of Contents

Acknowledgements.................................................................................................................. 5

Biographical Note..................................................................................................................... 7

Table of Contents...................................................................................................................... 9

Table of Figures.......................................................................................................................... 11

Table of Tables.......................................................................................................................... 12

1. INTRODUCTION............................................................................................................. 13
   1.1 Problem Statement ......................................................................................................... 13
   1.2 Thesis Overview ............................................................................................................ 14

2. AMAZON.COM AND CUSTOMER SERVICE ................................................................... 15
   2.1 History of Amazon.com ............................................................................................... 15
   2.2 Customer Service at Amazon.com .............................................................................. 16
   2.3 CSA Workflow ............................................................................................................. 19
   2.4 CSA Metrics ............................................................................................................... 20

3. CONTACT ROOT CAUSE ANALYSIS........................................................................... 21
   3.1 Relevant Business Teams ............................................................................................ 21
   3.2 Classification of Concessions ...................................................................................... 21
   3.3 Classification of Contacts ........................................................................................... 24

4. APPLYING LEAN THINKING TO CUSTOMER SERVICE .............................................. 27
   4.1 Lean Thinking ............................................................................................................ 27
   4.2 Applying Lean Thinking to Consumption .................................................................. 29
   4.3 Applying Lean Thinking to the Coding Process ....................................................... 30

5. DEVELOPING NEW REASON CODES........................................................................ 33
   5.1 Understanding the Needs of the Customer ............................................................... 33
   5.2 Mapping Decisions to Codes ................................................................................... 35
Table of Figures

Figure 1: American Customer Satisfaction Index for Internet Retail Industry.......................... 16
Figure 2: Web form for e-mail submission.............................................................................. 17
Figure 3: Value stream map for order-related phone contact .................................................. 20
Figure 4: Screenshot of Reason Code Decision Tool .............................................................. 38
Figure 5: Redesigned Web Form ............................................................................................ 43
Table of Tables

Table 1: Drop-down menu issue options by product category .................................................. 18
Table 2: Reason codes for concessions .................................................................................... 23
Table 3: Comparison of current to updated reason codes .......................................................... 34
Table 4: AHT reduction assumptions ....................................................................................... 44
Table 5: Summary of potential savings by collecting customer information prior to contact .... 46
1. INTRODUCTION

The following thesis is based on a six-month internship that was conducted at the corporate headquarters of Amazon.com in Seattle, WA from February 2008 to August 2008. The internship and following thesis are requirements for the Leaders for Manufacturing program at the Massachusetts Institute of Technology.

In this chapter, I will introduce the problem and provide an overview of how the thesis is organized.

1.1 Problem Statement

When a customer calls or e-mails Amazon customer service, the customer service agent (CSA) will diagnose the issue, render a solution, and then wrap-up the call or e-mail. This entire process is done on an internally developed computer program. The wrap-up portion requires the CSA to choose, from a list of radio buttons, the reason the customer contacted Amazon customer service. Within the customer service (CS) department, this reason is referred to as the issue code. Additionally, if a contact requires a concession (free replacement, refund, shipping charge reduction, etc.), there is an additional workflow that the CSA must complete prior to wrap-up. This workflow requires the CSA to choose, from a list of radio buttons, the reason the customer requires a concession.

As of the time of research, the issue and reason codes do not drive down to the root cause for a contact or concession, respectively. For example, one of the most widely used issue codes is “Item Issue: Post-Order,” but what is the post-order issue? Additionally, these codes have not been updated to reflect the changing needs of the business (i.e. there exist no reason codes for new product offerings such as MP3 downloads, video downloads, etc.). Making matters worse, the CSA has neither the time nor the motivation to choose the issue code (and reason code, if necessary) that best fits the situation. As a result, CSA miscoding is a common occurrence. This miscoding, coupled with codes that are not actionable, make it difficult for the relevant business team (retail, transportation, fulfillment center, etc.) to use the coding information provided by CS.
1.2 Thesis Overview

The following is an overview of the thesis, highlighting the main topics of each chapter.

Chapter 1: This chapter outlines the problem of the company’s inability to efficiently identify and correct the root cause of a customer contacting customer service, and provides an outline of the thesis.

Chapter 2: This chapter covers a brief history of Amazon.com and discusses the conduct of customer service at Amazon.com.

Chapter 3: This chapter describes the process of analyzing the root cause behind a customer contact.

Chapter 4: This chapter outlines the principles of lean thinking and how they can be applied to customer service.

Chapter 5: This chapter describes how lean principles were applied to develop new codes for concessions.

Chapter 6: This chapter describes how lean principles were applied to standardizing the work performed by CSAs.

Chapter 7: This chapter describes how lean principles were applied to standardizing the work performed by customers.

Chapter 8: This chapter concludes the thesis and provides recommendations.
2. AMAZON.COM AND CUSTOMER SERVICE

2.1 History of Amazon.com

Amazon.com, a Fortune 500 company, was founded by Jeff Bezos in 1994 and is headquartered in Seattle, WA. Originally started as an online bookstore, the company now serves three primary customer groups: consumer customers, seller customers, and developer customers.

Consumer customers are served through the company’s retail websites. “[Amazon] endeavor[s] to provide the widest possible selection for customers worldwide, while continuing to focus on in-stock inventory availability” (Amazon.com Inc., 2008). Consumer products are organized into the following categories:

- Books
- Electronics & Computers
- Toys, Kids & Baby
- Sports & Outdoors
- Movies, Music & Games
- Home & Garden
- Apparel, Shoes & Jewelry
- Tools, Auto & Industrial
- Digital Downloads—this includes MP3s, videos, and electronic books (for use on Amazon’s electronic book reader, Kindle)
- Grocery
- Health & Beauty

Seller customers are served by Amazon Services, which allows individuals, small businesses, and large businesses to sell products on the company’s retail websites.

Developer customers are served by Amazon Web Services, which provides access to web-based services such as data storage and cloud computing.
The paper is focused on the consumer business of Amazon.com.

2.2 Customer Service at Amazon.com

According to the American Customer Satisfaction Index, Amazon.com has been one of the highest rated companies amongst customers (The American Customer Satisfaction Index, 2009). Figure 1 contains the graph of the American Customer Satisfaction Index for the internet retail industry since 2000. A major contributing factor to Amazon’s outstanding performance is its superior customer service.

![American Customer Satisfaction Index for Internet Retail Industry](image)

Figure 1: American Customer Satisfaction Index for Internet Retail Industry

A customer of Amazon.com has three options for handling issues: Call customer service, e-mail customer service, or self-serve.

If a customer wants to talk to a live CSA, they can either call a 1-800 number or be called by using the company’s click-to-call feature. This feature allows a customer to input their number...
via the website and indicate when they would like to be called. The customer will then be contacted by a live agent. This feature prevents the customer from having to wait in a queue while on the line. Additionally, this feature reduces the time required for the CSA to perform the security verification check since the customer’s personal information is tied to the phone number.

A customer can also send an e-mail through a form provided on the website, as shown in Figure 2. The customer will provide an order number (if applicable), select an issue from a drop-down menu, provide additional free-form text, and submit the e-mail.

![Figure 2: Web form for e-mail submission](image)

The issue options in the drop-down menu depend on whether the e-mail is in regards to an MP3 issue, a video issue, a Kindle issue, or a general issue. Table 1 contains a summary of the issue options in the drop-down menu. The option selected by the customer will become the subject line for the e-mail when viewed by the CSA. Additionally, the e-mail can be routed based on the issue option selected.
<table>
<thead>
<tr>
<th>MP3</th>
<th>Video</th>
<th>Kindle</th>
<th>General</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3 Downloads: Wrong Song</td>
<td>Video Availability</td>
<td>Kindle Book Questions</td>
<td>Where's My Stuff?</td>
</tr>
<tr>
<td>MP3 Downloads: Downloading Issues</td>
<td>Feedback</td>
<td>New Kindle User Questions</td>
<td>Placing an Order</td>
</tr>
<tr>
<td>MP3 Downloads: Missing or Corrupt Audio</td>
<td>New User Questions</td>
<td>Kindle Book Availability</td>
<td>Gift Certificates/Cards &amp; Promotional Offers</td>
</tr>
<tr>
<td>MP3 Downloads: Downloader Application Issues</td>
<td>Ordering Problems</td>
<td>Publish Kindle Content</td>
<td>Returns &amp; Refunds</td>
</tr>
<tr>
<td>MP3: PepsiStuff Points and MP3 Orders</td>
<td>Player Issues</td>
<td>Feedback</td>
<td>Received Damaged, Defective, or Wrong Item</td>
</tr>
<tr>
<td>Other Questions &amp; Comments</td>
<td>Problems Viewing On Your TV</td>
<td>Wireless Connectivity Issues</td>
<td>MP3 Downloads</td>
</tr>
<tr>
<td></td>
<td>Problems Streaming</td>
<td>Kindle Reader Issues</td>
<td>Corporate Orders &amp; Accounts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Damaged/Defective)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kindle Subscriptions</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Personal Document Issues</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Drop-down menu issue options by product category

Skill-based routing at Amazon directs e-mails to specific queues based on both the issue option selected and keywords in the free-form text box. CSAs must have the appropriate skill to access a specific queue. This system ensures that e-mails are handled by capable CSAs.

Self-service is provided through three avenues: the website’s help pages, the website’s Express Customer Service tool, and the interactive voice response (IVR) system via the 1-800 number. The help pages allow the customer to search for information by topic. The Express Customer Service tool allows the customer to take order-specific actions, such as package tracking, returns, etc. Finally, a customer contacting Amazon through the 1-800 number is able to serve themselves for limited issues via an IVR system.
2.3 CSA Workflow

The CSA workflow will vary depending on whether the contact is made via phone or e-mail, whether the phone contact is click-to-call, and whether the contact is related to an order or not.

For a phone contact, the following steps are taken by the CSA:

- Greeting/Security Check
  - Security check takes longer for 1-800 phone contacts.
- Pre-Diagnosis
  - Listens to customer’s issue/question.
- Research
  - This phase takes longer for non-order related contacts.
- Transfer
  - This phase is performed if the CSA does not have the appropriate skills to handle the contact.
- Diagnosis
- Solution
  - Selects reason code indicating reason for concession, if necessary.
- End Call
- After Call/Wrap-up
  - Submits web forms, writes follow-up e-mails, etc.
  - Sends a customer feedback form, referred to as a How’s My Driving? (HMD) survey.
  - Selects issue code indicating reason for contact.

For an e-mail contact, the following steps are taken by the CSA:

- Pre-Diagnosis
- Research
- Take Action
  - Selects reason code indicating reason for concession, if necessary.
- Respond to E-mail
• Wrap-up
  o Sends an HMD survey.
  o Selects issue code indicating reason for contact.

Figure 3 contains a value stream map for an order-related phone contact.

![Value stream map for order-related phone contact]

**2.4 CSA Metrics**

CSAs are evaluated on a number of metrics:

- Contacts Handled per Labor Hour
- Expressed Dissatisfaction Rate (EDR)
  - Calculated by dividing the number of HMD “No” responses by the number of HMD offerings.

The goal for a CSA is to maximize contacts per labor hour while minimizing EDR. There exists no metric that evaluates the accuracy with which a CSA chooses a reason code or issue code. With no metric tracking issue code and reason code accuracy, there is little incentive for the CSA to “get it right.”
3. CONTACT ROOT CAUSE ANALYSIS

This chapter will explain the business teams that are impacted by the CSA coding process, as well as the methods Amazon currently employs to classify the reason for a customer contact and the reason for a concession granted to a customer.

3.1 Relevant Business Teams

At Amazon, the warehouses that store, process, and package the consumer goods are known as fulfillment centers (FCs). The FC business team is responsible for ensuring that consumer goods are accurately and safely packaged for the right customer in a timely manner.

The retail teams are responsible for sourcing consumer goods from the supplier to the FC. Additionally, they are responsible for ensuring the website reflects the correct information regarding the particular consumer good.

The transportation team is responsible for coordinating with the contracted shipping companies (e.g. UPS, FedEx, etc.) to ensure that packages are safely shipped from the FC to the right customer in a timely manner.

The customer service (CS) team is responsible for conducting call center operations and coordinating with the above business teams regarding concessions.

3.2 Classification of Concessions

When a CSA is taking an action during a call or e-mail that requires a concession, the CSA must classify the reason for a concession by selecting a reason code from a list. Table 2 contains a list of current reason codes. Reason codes are used to understand the root cause for a granted concession, and help assign responsibility for the concession. For example, if a customer calls because they received the wrong item, the CSA will either send a free replacement or issue a refund, and then select the most appropriate reason code. In this case, the CSA will select “Wrong Item Received (Switcheroo)” and the FC will be assigned fiscal responsibility for the granted concession.
When the wrong item is received, the responsible party is clear. However, if an item arrives damaged, who is responsible? It could be the item was damaged by the FC during the warehouse processing and packaging phase. Or the item could have been damaged during the shipping process. Appropriately assigning responsibility (by choosing the correct reason code) would require some deeper level analysis. Was the outer box damaged or not? If so, then the transportation team should be held fiscally responsible, otherwise the FC team should be held responsible. Unfortunately, the CSA does not have a standardized list of questions to ask the customer to aid in code selection. Further, even if a standardized question list did exist, it would only be feasible for use with live, phone contacts.

<table>
<thead>
<tr>
<th>Concession Type</th>
<th>Reason Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Free Replacement</td>
<td>CS Slow to Respond</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Customer Error</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Missed EDD</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Defective Item in Shipment</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Item Damaged by Shipper</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Shipment Appears Lost but not past EDD</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Item Damaged due to Poor Packaging</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Item Damaged (Undetermined Cause)</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Shipment in Cleanup</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Delivery Problem, Address Correct</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Wrong Item Received (Switcheroo)</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Item Damaged in FC</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>ESD/EDD Impossible</td>
</tr>
<tr>
<td>Free Replacement</td>
<td>Item Missing from Shipment</td>
</tr>
<tr>
<td>Refund</td>
<td>Advance refund for return</td>
</tr>
<tr>
<td>Refund</td>
<td>Returns-related error</td>
</tr>
<tr>
<td>Refund</td>
<td>Unwanted item/could not cancel</td>
</tr>
<tr>
<td>Refund</td>
<td>Customer Error</td>
</tr>
<tr>
<td>Refund</td>
<td>Item Does Not Meet Customer Expectations (Incomplete Catalog Info)</td>
</tr>
<tr>
<td>Refund</td>
<td>Discount for Damaged but kept item</td>
</tr>
<tr>
<td>Refund</td>
<td>Defective Item in Shipment</td>
</tr>
<tr>
<td>Refund</td>
<td>Shipment Appears Lost but not past EDD</td>
</tr>
<tr>
<td>Refund</td>
<td>Item Damaged due to poor packaging</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Refund</td>
<td>Item received materially different (catalog error)</td>
</tr>
<tr>
<td>Refund</td>
<td>Item not compatible (Incomplete catalog info)</td>
</tr>
<tr>
<td>Refund</td>
<td>Delivery problem, address correct</td>
</tr>
<tr>
<td>Refund</td>
<td>Item damaged in FC</td>
</tr>
<tr>
<td>Refund</td>
<td>Item Missing from Shipment</td>
</tr>
<tr>
<td>Refund</td>
<td>Customer ordered wrong item</td>
</tr>
<tr>
<td>Refund</td>
<td>Retroactive discount</td>
</tr>
<tr>
<td>Refund</td>
<td>Item damaged by shipper</td>
</tr>
<tr>
<td>Refund</td>
<td>Item damaged (undetermined cause)</td>
</tr>
<tr>
<td>Refund</td>
<td>Wrong item received (switcheroo)</td>
</tr>
<tr>
<td>Refund</td>
<td>CS Slow to Respond</td>
</tr>
<tr>
<td>Refund</td>
<td>Missed EDD</td>
</tr>
<tr>
<td>Refund</td>
<td>Billing error</td>
</tr>
<tr>
<td>Refund</td>
<td>Found better price elsewhere</td>
</tr>
<tr>
<td>Refund</td>
<td>Promotion/marketing error</td>
</tr>
<tr>
<td>Refund</td>
<td>Pricing/content error</td>
</tr>
<tr>
<td>Refund</td>
<td>Other price reduction</td>
</tr>
</tbody>
</table>

**Table 2: Reason codes for concessions**

In some cases, the responsible party is not ambiguous, but the CSA simply chooses the wrong reason code and thus improperly assigns fiscal responsibility. While this situation causes conflict between the CS team and the misclassified team, the CSA has little incentive to improve coding accuracy, especially at the expense of reducing the number of contacts handled per labor hour.

There are other cases where the responsible party is not ambiguous, and the CSA chooses the right reason code, but the reason code does not help the responsible team understand what went wrong. For example, if a customer orders a television with a remote included, but does not receive the remote, then the CSA should select the reason code “Defective Item in Shipment,” and fiscal responsibility will be assigned to the appropriate retail team (i.e. consumer
electronics). However, the consumer electronics team has no insight into how the item was defective without having to devote research time to investigate.

### 3.3 Classification of Contacts

After a call or e-mail, the CSA will classify the reason for a contact by selecting the most appropriate issue code from a list. Unlike reason codes, issue codes do not have a responsible team assigned to them as there are no direct fiscal implications for a customer contact. Instead, the issue codes are used by the CS team to track the frequency of code usage. This helps the CS team to detect any trends in the fulfillment process. Below is a list of the current issue codes:

- Account Maintenance
- Item issue: Post-Order
- Order Revision: Post-Order
- Password Reset/Security Question
- Payment and Billing Activity: Post-Order
- Positive Feedback
- Pre-Order
- Problem Transactions or Guarantee Issues: MP
- Promotions
- Returns and Related Refunds
- Transfer/Refer Customer or Incomplete Contacts
- Troubleshooting
- Troubleshooting—Digital Hardware Troubleshooting
- Troubleshooting—Digital Online Services
- Troubleshooting—Digital Software Troubleshooting
- Website and Company Info
- WMS: Late by FC
- WMS: Late by Shipper
- WMS: Late by Supply Chain
- WMS: On Time
- Pepsi: Account Maintenance
• Pepsi: Item Issue: Post-Order
• Pepsi: MP3 Download Issue
• Pepsi: MP3 Troubleshooting
• Pepsi: Password Reset/Security Question
• Pepsi: Payment and Billing Activity: Post-Order
• Pepsi: Point addition/deletion
• Pepsi: Promotions/Sweepstakes
• Pepsi: Returns and Related Refunds
• Pepsi: Transfer to Digital Team
• Pepsi: Transfer/Refer to Pepsi
• Pepsi: Unbox Issue
• Pepsi: Website: pepsistuff.com
• Pepsi: WMS: Late by FC
• Pepsi: WMS: Late by Shipper
• Pepsi: WMS: Late by Supply Chain
• Pepsi: WMS: On-time

Note: The “Pepsi” codes refer to issues that arise over Amazon’s promotional offer with Pepsi where Pepsi customers can collect points to purchase digital content (i.e. music, movies, etc.) on Amazon.com

Currently, the issue code selection process takes anywhere from 2% to 4% of a CSA’s time while processing a contact (based on live observation).
This page has been intentionally left blank
4. APPLYING LEAN THINKING TO CUSTOMER SERVICE

Lean thinking has been evangelized by James Womack and Daniel Jones since the 1996 release of their book, *Lean Thinking*. The book was published to answer the questions that arose after Womack and Jones (along with Roos) wrote *The Machine That Changed the World* in 1990. *The Machine That Changed the World* was based on a five-year study of the automotive industry, which found that Toyota, a Japanese auto manufacturer, had developed a more efficient way to manufacture automobiles through their Toyota Production System (TPS). *Lean Thinking* outlined methods for an organization to eliminate waste during production (Womack, Jones, & Roos, 2007).

The concept of lean production was most easily adopted by manufacturing companies. Could the principles of lean production be applied to service companies as well? In 2005, Womack and Jones wrote *Lean Solutions*, a book that developed the idea of lean consumption, “a better way for consumers to obtain the goods and services they now want” (Womack & Jones, 2005). The concept of lean consumption can be applied to both manufacturing and service companies.

4.1 Lean Thinking

Lean thinking is “a way to specify value, line up value-creating actions in the best sequence, conduct these activities without interruption whenever someone requests them, and perform them more and more effectively” (Womack & Jones, 2003). More explicitly, lean thinking is applied by:

- Specifying the value that a process is creating.
- Identifying the value stream and removing steps that are wasteful.
- Placing the value-creating steps into continuous flow.
- Allowing the customer to “pull” value from the stream.
- Continuously improve the process in pursuit of perfection.

Before one can improve a process through lean principles, they must specify the value that is created by a process. The process of brewing a pot of coffee produces value by generating a
caffeinated beverage that keeps the consumer mentally alert. By knowing exactly what value is created, you begin to understand the reason for the various steps in the process.

Once the value has been specified, one must identify the value stream, which is the set of actions that are required to produce an item. In the case of a cup of coffee, the value stream might include placing a coffee filter in the basket, filling the filter with coffee grounds, inserting the basket into the coffeemaker, placing a carafe under the basket, and pouring water into the coffeemaker.

After the value stream has been identified, the goal is to organize the steps into a continuous flow, meaning water is poured into the coffeemaker only after the carafe has been placed under the basket, and the carafe is placed under the basket only after the basket has been inserted into the coffeemaker, etc. The converse of continuous flow is batch-and-queue, where steps are not done in order, and batches of products after one step build up in queues in front of the other steps.

Once steps are organized into a continuous flow, one can utilize a “pull” system where the removal of a product at the end of the value stream triggers the previous step in the stream. Using this method, products are manufactured as they are demanded. The converse of a pull system is a push system where products are built according to a forecast. This leads to a large finished goods inventory that costs money and has the potential for overproduction.

The final principle in lean thinking is to continuously improve upon every step in your process. Continuous improvement leads to a mindset where the current state of a process is subpar.

These principles were developed for use in manufacturing processes where physical products are produced. The concept of lean proved so successful for manufacturing companies, that many wondered if the principles could be applied outside of manufacturing.
4.2 Applying Lean Thinking to Consumption

Womack and Jones applied the concept of lean thinking to the act of consumption. They argue that people solve their problems through the consumption of either products and/or services. A lean consumption process will follow six principles (in the voice of the customer) (Womack & Jones, 2005):

- Solve my problem completely.
- Don’t waste my time (minimize my total cost of consumption, which is the price I pay plus my time and hassle).
- Provide exactly what I want.
- Deliver value where I want it.
- Supply the value when I want it.
- Reduce the number of decisions I must make to solve my problems.

In Lean Solutions, Womack and Jones describe a typical problem that is solved through consumption: using a word processor to create a document by purchasing a computer (Womack & Jones, 2005). A customer will begin this process by researching product information to determine the best computer to purchase. Once the purchase decision is made, the customer goes to the manufacturer’s website and makes the purchase. After a few days, the computer arrives via a shipper. The computer does not come loaded with the correct software for word processing, however, so the customer makes an additional software purchase and performs an install. After the install, however, the computer no longer starts up. The customer makes a call to the computer manufacturer, and after wasting time on the phone, learns that the issue is with the software, not the hardware. This prompts another call by the customer to the software company, and after additional wasted time, the customer learns that there is nothing wrong with the software. The customer finally calls a computer technician who comes in person to fix the issue and allow the customer to create a word processor document.

According to Womack and Jones, this consumption process is not lean because it does not follow the six principles stated above. To improve the consumption process described above it must follow these six principles. In order to solve the customer’s problem completely, the computer
would have been pre-loaded with the appropriate word processing software. This would have prevented the customer from wasting their time (because they would not be forced to purchase additional software) and prevented them from calling customer service when their computer crashed. Purchasing a computer with pre-loaded word processing software provides the customer with exactly what they want (i.e. a means to produce a document), delivers value where they want it (i.e. within a single program that is pre-loaded on the computer they purchased), supplies value when they want it (i.e. at the time of computer purchase and no later), and reduces the number of decisions they must make to solve their problem (i.e. they only have one decision to make: whether or not to purchase the computer).

**4.3 Applying Lean Thinking to the Coding Process**

The entity that begins the process of correcting the root cause of a granted concession is the reason code selected by the CSA. Therefore, the reason coding process, performed by the CSA, became the focus for applying the principles of lean consumption. My hope was to apply any lessons learned from the concession coding workflow to the issue coding workflow.

Applying the principles of lean consumption to reason coding requires a definition of who the customer is. In the complex space of reason coding, there are many customers with differing roles:

- The business team who uses the reason code to improve their business practices.
- The CSA who analyzes the information provided by the customer and selects the most appropriate reason code.
- The Amazon.com customer who provides the information required to select a reason code.

I first looked at the business team, as a customer through the lens of lean consumption, to understand areas for improvement in the coding process. After a concession is granted, the relevant business team has the problem of understanding why the concession was granted. The reason code is supposed to solve this problem; however, the current codes are not descriptive enough to solve the customer’s problem completely. Informing a particular retail team that a
consumer received a free replacement due to a “Defective Item in Shipment” only partially solves the customer’s problem. In order to fully solve the problem, the business team must conduct research to understand the root cause for a concession. This process, requiring extra man-hours, produces no value, and wastes the time of an employee who should be creating value. It becomes clear that in order to solve the customer’s problem completely, there must be a more robust list of codes that provide the detailed reason for a concession. Chapter 5 discusses the efforts taken to apply lean thinking to the needs of business team.

Looking at the CSA as a customer, we begin to see areas of improvement for the coding workflow. The CSA’s problem is how to pick the most appropriate reason code for a granted concession. Currently, the CSA does not know whether or not to ask questions to determine the reason for a concession, and if they do decide to ask questions, they do not know which questions to ask. When choosing a reason code, the CSA is presented with an overwhelming list of codes. This process does not solve the customer’s problem completely; it wastes the time of a CSA who could be helping other customers; and it does not reduce the number of decisions that need to be made in order to solve the problem. A better solution will tell the CSA which questions to ask and when to ask them, with the final outcome being a chosen reason code, not a screen filled with myriad options. Chapter 6 discusses the efforts taken to apply lean thinking to the CSA coding workflow.

Finally, looking at the Amazon.com customer, the fulfillment process has already failed based on the requirement of a concession. Referring to the customer service value stream map in Figure 3, the coding process provides no value for the customer. Further adding to waste is the customer being subjected to a varying string of questions depending on the amount of research the CSA decides to conduct for reason code determination. A standardized list of questions that is asked of the customer might still provide no value for the customer, but it will correct the systemic failure that occurred in the fulfillment process. Another time-wasting activity for the customer is the paraphrasing and clarifying questions that must be asked by the CSA to reconnoiter the customer’s situation. What if the CSA already knew why the customer was calling, and didn’t have to waste the customer’s time with additional clarifying questions? Or questions related to code determination? I wanted to explore the possibility of removing or reducing these time
wasting activities while still gaining accurate and meaningful insight about granted concessions. Chapter 7 discusses the efforts taken to apply lean thinking to the customer workflow.

I used the three perspectives above to generate ideas for improving the coding process. In the next three chapters, I will outline these initiatives.
5. DEVELOPING NEW REASON CODES

As described in the last chapter, from the perspective of the relevant business team as the customer, the current concession codes do not completely solve the customer’s problem of determining the root cause for a granted concession. To remedy this shortfall, I interviewed members of the various business teams to develop a new list of codes. Once I had a thorough list of codes, I wanted to map the codes as outcomes of binary or near-binary questions.

5.1 Understanding the Needs of the Customer

I met with the following business teams to develop a new list of reason codes that would help the team determine the root cause for a granted concession:

- Customer service (CS)
- Fulfillment center (FC)
- Retail
  - Automotive and Industrial
  - Books
  - Consumables
  - Consumer Electronics
  - Home and Garden
  - Jewelry and Apparel
  - Music and Movies
  - Shoes
  - Sports and Home Improvement
  - Toys
  - Video Games and Software
  - Transportation

In these meetings, I wanted to understand the additional information the particular business team would want in order to correct the root cause for a customer contact. For example, if a customer contacted customer service because of a defective toy they purchased, the retail team responsible for toys would want to know the following:
The retail team responsible for toys wanted this information in order to effectively work with their vendors to correct the underlying issue of a defective product being shipped to customers.

With these interviews completed, I developed a new list of reason codes for concessions. Table 3 contains a comparison of current to updated reason codes for two current reason codes: defective item in shipment and item damaged (undetermined cause).

<table>
<thead>
<tr>
<th>Current Reason Code</th>
<th>Updated Reason Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Missing pages</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Misprinted pages</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Incorrect binding</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Defective binding</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Incorrect edition</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: No/incorrect add-ins</td>
</tr>
<tr>
<td>Defective item in shipment</td>
<td>Defective item in shipment: Other</td>
</tr>
<tr>
<td>Item damaged (undetermined cause)</td>
<td>Item damaged: No boxes damaged</td>
</tr>
<tr>
<td>Item damaged (undetermined cause)</td>
<td>Item damaged: Outer shipping box damaged</td>
</tr>
<tr>
<td>Item damaged (undetermined cause)</td>
<td>Item damaged: Inner retail box damaged</td>
</tr>
<tr>
<td>Item damaged (undetermined cause)</td>
<td>Item damaged: Both boxes damaged</td>
</tr>
</tbody>
</table>

Table 3: Comparison of current to updated reason codes

Using the current list of reason codes, if a customer were to receive a book with missing pages, the correct reason code to be selected, if a concession was granted, would be “Defective Item in Shipment.” The retail team responsible for books would not have any insight into the defective
item based on this code. Using the new list, the correct reason code would be “Defective Item in
Shipment: Missing Pages,” and the retail team would now have a better understanding of the
failure in the fulfillment process, allowing for efficient correction. As a result of this research,
the new list of concession codes expanded more than three fold.

It was clear during this process that aside from codes that did not clearly indicate where the
failure occurred in the fulfillment process, there was also an issue of overlapping, or ambiguous,
responsibility for a failure. I wanted the new list of codes to give a clear indication of
responsibility. Additionally, as a result of the interviews, I began to understand the questions
that needed to be asked in order to make a code determination. With the help of the business
teams and the CSAs, I mapped decisions to the new codes.

5.2 Mapping Decisions to Codes

The decision map I developed used yes/no and limited multiple choice questions to reach an
ultimate conclusion, or reason code. An example string of questions might be:

- Was a shipment received? (Yes/No)
  - Yes
- Did shipment arrive late? (Yes/No)
  - No
- Is outer box damaged? (Yes/No/Unknown)
  - Yes
- Conclusion: Item damaged by shipper (Transportation responsibility)

Aside from verifying the correct decisions were mapped to the correct codes by consulting with
the business teams, I also ensured that the questions were worded using the language recognized
by the CSAs. As an example, a missing item is different than a missing bundled item. I wanted
to ensure that none of the questions used language that was confusing for the CSA.

The hardest parts of the mapping process were eliminating the answer of “unknown,” and
limiting the answer possibilities to a simple yes or no. The reality, however, is that there will
always be customer service situations where “unknown” is the most appropriate answer. For
example, for a damaged item, if a customer cannot remember the state of the packaging, then a determination of how it was damaged is unable to be made. As for limiting the answer possibilities, there were certain circumstances where a clear yes or no answer was not possible and thus required multiple choices. I tried to eliminate questions that had ambiguity, but there remained decisions that were not based on a yes or no answer. As an example, there is one string of questions that asks whether an item was missing, wrong, or otherwise flawed. On top of having three choices for this question, if the customer says “otherwise flawed,” the next question the CSA must answer is which product category the item belongs to. Not only are there many product categories, but the customer (and sometimes the CSA) is not going to know the correct product category. Additionally, many items could fall into multiple product categories, making it difficult for the CSA to choose the correct category. Ideally, this is a decision that should not have to be made by a human, but this would require software resources to remedy. With the exception of a few circumstances, however, I limited most decisions to a simple yes or no.

With a decision tree that mapped to the new codes I developed, the next step in the process was to develop the tree into a tool that could be used by the CSA. In the next chapter, I describe how I developed this tool and how it was used to standardize the coding process for the CSA.
6. STANDARDIZING WORK FOR CSAS

This chapter examines the feasibility of standardizing the coding workflow for the CSA. Standardizing this workflow involves having the CSA answer a set of standardized questions to reach a conclusion (i.e. issue code or reason code). Since there are two coding workflows (reason coding and issue coding), I decided to focus the standardizing effort on the reason coding workflow, as there were fiscal implications for reason miscoding, and because the new codes I developed with the business teams were for concessions.

I used the new list of codes I developed, and the corresponding decision tree, to create a prototype tool that the CSA could use in the coding workflow. My goal in developing this tool was to assess if the new list of codes and decision trees was an improvement over the current state. I performed this assessment by employing the prototype tool in a pilot test with CSAs in the U.S. and India. With the test complete, I tried to leverage existing software to develop a working tool for the CSA. I discuss the implementation issues that hindered this process.

6.1 Reason Coding Decision Tool

The issue with the current coding process is that it is not standardized. The CSAs either do not know, or do not care to ask the right questions to understand the root cause of a granted concession in order to properly choose a reason code. As a result, the same questions do not get asked of the customer to determine where a failure in the fulfillment process occurred. For example, if a customer calls regarding a damaged item, the CSA might or might not ask if the outer shipping box was damaged to determine where in the process the item was damaged. Instead, the CSA has a reason code that would be correct, but not helpful: “Item Damaged (undetermined cause).”

Armed with a new list of reason codes and a decision tree mapping decisions to codes, I developed a rudimentary tool to begin standardizing the coding process. This tool used HTML and JavaScript to create expanding menu lists based on the prior selection. A screenshot of the tool is contained in Figure 4. The benefit of this tool was that an CSA could quickly click back and forth through the menus, allowing them to change their decisions based on new information.
When a conclusion was reached, a reason code was presented, as well as a “submit” button to allow the CSA to enter the code into the system.

**Reason Code Guide**

*Why is the customer contacting us?*

- No receipt
- Post receipt
- Return related
- Pricing/billing related
- Did package arrive late?
- Is item damaged, defective/wrong/
  missing, or unwanted?
- Yes
- No

![Submit](Submit)

Clear All

**Figure 4: Screenshot of Reason Code Decision Tool**

During the CSA workflow, the CSA performs multiple functions simultaneously by having many windows open on their desktop screen. The vision was for the CSA to have an additional window open for the above tool that could be used during the phone call or e-mail. The main limitation of this tool was that it could not be used properly if the CSA could not answer all the questions. This was a major limitation for e-mail contacts where the customer often did not provide the requisite information that could otherwise be asked if the customer was a live phone contact.

**6.2 Pilot Test**

I conducted a pilot test with around 150 CSAs at four call centers to determine if standardizing the reason coding workflow could improve coding accuracy. To accomplish this, I developed 20 customer scenarios that explained a particular problem. I ensured that all required information to code the contact was present in the scenario. While this did not reflect real life, I wanted this test to be a first pass to determine if the decision tool path was worth investing further efforts.
Once I developed the 20 scenarios, I separated the participants into two groups. The first group would read the scenario and choose a code as they normally would. The second group would read the scenario and use the coding tool to choose a code. I consulted with CSAs and member of all the business teams to ensure the “correct” answer for each scenario.

I conducted the test by posting the scenarios and the tool to an internal web page. The coding answers were transmitted to an internal database where I performed a SQL query to gather the data into Excel. Of the 139 participants that selected codes without the tool, 59.9% selected the “correct” code. Of the 137 participants that selected codes with the tool, 70.6% selected the “correct” code. Additionally, by looking at the accuracy rates of specific questions, I was able to further refine the decision tree to eliminate ambiguity. To determine if the difference in test results was statistically significant, I used a two-tailed hypothesis test with a 0.10 significance level. I calculated a P-value of 0.06; therefore I rejected the null hypothesis and concluded that the difference in means was significant. As a result, I decided that the tool was worth devoting further resources. Fortunately, Amazon had an internally developed decision software tool known as the “wizard.”

6.3 CSC Wizard

Customer Service Central (CSC) is Amazon’s internally developed software program used by their CSAs to handle contacts. There exists a software tool within CSC, known as the “wizard,” that uses both manual CSA inputs and automatic inputs (e.g. tracking information, etc.) to reach a conclusion. It was believed that this tool could be leveraged to standardize the coding workflow and, by using the decision tree I developed for reason coding, aid the CSA in choosing the most appropriate reason code. The tool had limitations, however, that made it ineffective for use in the CSA workflow.

One limitation for this tool was the same limitation for the tool I developed: it required certain, specific information that the customer did not or could not always provide. Another limitation, was that the wizard was optimized for automated decisions (decisions that were based on existing data such as tracking information) and thus ran slowly when there were too many manual decisions (decisions that required input from the customer or required the judgment of a
Based on page load times, and the number of manual decisions, processing time alone ranged from 45-55 seconds. A wizard plan for reason coding was developed and tested in production, but this performance issue prevented the wizard from effectively being used to help the CSA choose the most appropriate concession code. In other words, the CSA simply stopped using the tool and resorted to the current method of selecting the code based on their judgment. The CSA did not have the additional time to sit through a slow software program when they were being measured on the number of contacts they handled in an hour. A third limitation was that the wizard was not contained in a separate window to allow for simultaneous workflows. The CSA was required to leave their working screen every time they conducted the wizard test. Thus if new information was provided by the customer during a call that influenced the selection of a code, the CSA would have to re-enter the wizard workflow.

6.4 Next Steps

Developing a useful coding tool for the CSA is going to require either a revamp of the wizard software to remedy the slow performance issue, or development of an entirely new software program that will allow the CSA to perform the coding workflow simultaneously with the rest of their work. This software development requires massive coordination between many business teams and cannot be performed through the coordination of an intern and a single software developer. The operational excellence team that I worked with, and was developed to address work standardization issues, is the most appropriate coordinating device for these efforts.

Additionally, coordination with the data warehouse, to ensure that the new codes are inputted into Amazon’s data storage system, needs to be initiated.
7. STANDARDIZING WORK FOR CUSTOMERS

Faced with the limitations of the wizard, the final area that remained to be explored was the area of information provided by the customer. One company, Adams Apparel (pseudonym) was able to reduce e-mail traffic by 71% by requiring contacts to provide information prior to contact (Corporate Executive Board, 2006). Additionally, if customers were to provide more information prior to calling or submitting an e-mail, the CSA would be better equipped to choose the most appropriate code. If this customer information could then be stored, the wizard could be used to call on this stored data, make automated decisions, and select both a reason and an issue code. An idea was developed to create a system of dynamic drop-down menus that a customer would fill out prior to contacting customer service either via e-mail or through a click-to-call phone call.

Collecting information from the customer prior to contact, and using the wizard for coding, has the following potential benefits:

- Reduces contacts by surfacing self-service options (13 of the 108 issue codes can be self-serviced) for customers after they select their issue from the drop-down menus.
- Reduces CSA average handle time (AHT) by removing the coding workflow.
- Improves first-contact resolution (FCR) by reducing follow-up e-mails when insufficient information is provided by the customer.
- Reduces miscode research time spent by the business teams due to improved coding accuracy.

7.1 Redesigned Web Form

Currently, customers begin the customer service process by visiting the help pages on Amazon.com. The help pages contain links to self-service pages, as well as a “Contact Us” feature. When going to Contact Us, the customer has three options: Express, E-mail, and Phone. Under Express, there are targeted actions that can be taken for specific orders, ranging from changing the shipping address to canceling orders. The customer can also compose an e-mail, which requires selecting an issue from a drop-down list containing eight items, entering the order identification number (if applicable), and filling out a blank box with additional comments.
Finally, the customer can select the phone option, where they are presented with a box to enter their phone number for the click-to-call feature.

If the customer is currently coming to the help pages regardless of whether they are calling, submitting an e-mail, or helping themselves, there is an opportunity to further standardize the customer workflow when they first come to the help pages. Instead of having the customer decide whether their issue is best served with a phone, an e-mail, or self-help, a web form could be used to ask limited, but specific, questions of the customer. The conclusion of this web form would be a recommendation on the most appropriate channel for customer service.

I developed a web form that the customer would use to standardize their workflow. Figure 5 contains a screenshot of the redesigned web form. The customer would select from a maximum of four different dropdown menus, containing, at the most, seven items. After the customer selected from all the menus, a conclusion would be presented on what the most appropriate option for the customer is: phone, e-mail, or self-help. If the self-help option is the best one, a link to the correct page will be presented. The customer will still be able to choose the option they desire, but this will help customers make the best decision.
We answer most e-mails in less than 12 hours.

To: Amazon.com Customer Service
Account: Charlemagne Hottenstein, holyromanemperor@gmail.com
Order #: 103-6748159-9573022

Answer the following questions:

- An order I placed
- An order I received
- My order arrived damaged
- Outer shipping box damaged only

Provide any additional details

For security reasons, we strongly discourage the submission of credit card numbers through e-mail.

Figure 5: Redesigned Web Form

7.2 Benefits

This web form has the potential to reduce contacts by surfacing self-service options for customers. To determine the fiscal implications of reducing contacts, I first looked at the make-up of e-mail contacts by analyzing “blurbs.” Blurbs are pre-written responses that are copied and pasted into the reply e-mails that are sent from the CSA to the customer. There are blurbs that are used for issues that could be handled by the customer using self-service. By looking at the number of these blurbs that were used on an annual basis, I could get an idea of the number of customers with issues that could be self-serviced. With the current phone to e-mail mix being approximately equal, I made the assumption that for every e-mail customer with a self-service issue, there was a corresponding phone customer with a self-service issue. I then made the assumption that of these customers with self-service issues, 25% of them will actually serve themselves, and not contact customer service.
Aside from helping the customer decide the most appropriate channel for service, this web form has the potential to aid in the CSA coding workflow. By having the menu choices passed along to CSC, the wizard can call on the information and choose an appropriate code via automated decisions. With the wizard choosing the code, the coding workflow can be removed and thus reduce the CSA’s AHT. To determine the fiscal implication of reducing AHT, I first made some assumptions on how AHT would be reduced. Table 4 contains the assumptions I made. The pre-diagnosis work flow item constitutes the part of the phone call where the CSA asks clarifying questions to determine the customer’s issue. With information being provided by the customer, I made an assumption on the amount that this work flow item would be reduced by. Since the idea was for the web form to provide data to select both the issue and reason code, I assumed that both the concession coding work flow and the wrap-up (issue coding) work flow would be eliminated.

<table>
<thead>
<tr>
<th>Work flow item</th>
<th>Pre-Diagnosis (seconds)</th>
<th>Wrap-up (seconds)</th>
<th>Concession Coding (seconds)</th>
<th>Total (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Time to complete before Contact Us changes (sec)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail with order</td>
<td>60</td>
<td>10</td>
<td>10</td>
<td>80</td>
</tr>
<tr>
<td>E-mail without order</td>
<td>60</td>
<td>10</td>
<td>0</td>
<td>70</td>
</tr>
<tr>
<td>Phone with order</td>
<td>60</td>
<td>20</td>
<td>10</td>
<td>90</td>
</tr>
<tr>
<td>Phone without order</td>
<td>38</td>
<td>20</td>
<td>0</td>
<td>58</td>
</tr>
<tr>
<td><strong>Time to complete after Contact Us changes (sec)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pre-Diagnosis reduction assumption:</td>
<td>25%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail with order</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>E-mail without order</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>Phone with order</td>
<td>45</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>Phone without order</td>
<td>28.5</td>
<td>0</td>
<td>0</td>
<td>28.5</td>
</tr>
</tbody>
</table>

Table 4: AHT reduction assumptions

In order to complete the fiscal analysis, I also had to know the annual call volume, the ratio of phone contacts to e-mail contacts, the ratio of click-to-call contacts to 1-800 contacts (1-800 phone contacts had to be ignored since they are unaffected by this new treatment), the ratio of e-
mail pre-order to post-order contacts, the ratio of phone pre-order to post-order contacts, the variable cost per phone contact, the variable cost per e-mail contact, and the percentage of contacts that require a concession.

Another benefit involves increasing first-contact resolution (FCR) by reducing follow-up e-mails from the CSA when insufficient information is provided by the customer. To determine the fiscal implication of this, I looked at the total number of e-mail contacts in a year, the variable cost per e-mail contact, the percentage of e-mails that require a follow-up e-mail from the CSA due to a lack of information, and the average number of e-mails required to solve e-mails that have insufficient information (on average 1.5 e-mails). I then assumed that 10% of these follow-up e-mails would be reduced to determine the cost savings.

The final benefit that I analyzed to determine the fiscal impact was the reduction in miscode research time performed by the business teams due to increased coding accuracy. To determine this, I assumed a certain amount of time spent on each concession, the hourly rate for the employee conducting the research, and a 40% reduction in research time. I then looked at the total number of concession contacts to determine the savings derived from improving the accuracy of codes.

Based on the above benefits, Amazon.com has the potential to realize an annual savings of over $1.5 million, as shown in Table 5.
Table 5: Summary of potential savings by collecting customer information prior to contact

<table>
<thead>
<tr>
<th>Savings Opportunity</th>
<th>Assumptions</th>
<th>Potential Savings</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phone self-service reduction</td>
<td>0.2% of total contacts will use self-service ¹</td>
<td>$718K</td>
<td>45.5%</td>
</tr>
<tr>
<td>E-mail self-service reduction</td>
<td>0.2% of total contacts will use self-service ²</td>
<td>$166K</td>
<td>10.5%</td>
</tr>
<tr>
<td>AHT reduction</td>
<td>Remove wrap-up and reduce pre-diagnosis by 25%</td>
<td>$141K</td>
<td>8.9%</td>
</tr>
<tr>
<td>Improved FCR</td>
<td>10% reduction in unnecessary e-mails</td>
<td>$54K</td>
<td>3.4%</td>
</tr>
<tr>
<td>Miscode research time reduction</td>
<td>40% reduction</td>
<td>$499K</td>
<td>31.6%</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>$1,578K</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Due to data collection limitations, there are additional potential benefits of the web form treatment that have not been quantified:

- Reducing transfers by routing contacts to the most appropriately skilled CSA.
- Routing customers to the most appropriate channel (phone, e-mail, or self-service).
- Improving efficiency by providing business teams with information that will allow them to quickly attack the root cause of customer contacts.

### 7.3 Usability Test

Despite the potential benefits described above, a concern raised about the solution was that having the customer select from multiple menus would produce too much friction. By looking at the composition of current contacts, I determined that approximately 81% of customers will be presented with three menus or less, and approximately 79% of customers will be presented with two menus or less. Despite this information, I developed a usability test (in accordance with guidance from the Customer Experience team) using both the current Contact Us form, the future

¹ This represents a 25% reduction of customers who currently contact customer service for self-service issues
² This represents a 25% reduction of customers who currently contact customer service for self-service issues
Contact Us form, and four customer scenarios. I then had a group of participants respond to these scenarios using both the current form and the future form. The outcome of the test is summarized below:

- Total Participants: 20
- Participants from CS department: 11
- Participants not from CS department: 9
- 90% of those polled did not think the future form made it too difficult to contact customer service.
- 85% of those polled found it easier to express their issue/question in the future form.
- 80% of those polled preferred the future form.
- 5% of those polled had no preference between the current and future form.
- 60% of those polled wrote fewer additional comments in the future form.
- Participants correctly chose most appropriate issue 88% of the time.

In my opinion, the slight amount of friction created by the dynamic drop-down menu treatment can be justified by the cost benefits of collecting information from the customer prior to contact.

### 7.4 Next Steps

Despite the participants correctly choosing the most appropriate issue 88% of the time, further work needs to be done in refining the menu choices and ensuring that they are organized and worded in the most efficient manner. Once a dynamic drop-down menu version of the Contact Us page is developed, the goal should be to test the form with live customers in order to collect customer behavior information necessary to efficiently organize the menus. Ideally, customers will choose the most appropriate issue greater than 95% of the time.

Much of the work done on coding and customer service also has implications for the Online Return Center (ORC). This group handles the processing of returns made by the customer. The process of coordinating with ORC has already begun, but as more work is done, it is critical to ensure that this team is able to use the information collected from the customer via the new web form.
This page has been intentionally left blank
8. CONCLUSION

8.1 Recommendations

To effectively make use of issue and reason codes, two things are required: A list of codes that drives down to the root cause for a contact, and a method for either the agent or the customer to accurately choose a code.

Over the course of my internship, I developed a fairly robust list of codes that drove down to the root cause for a contact, based on input from various business teams. However, as Amazon.com continues to add more product lines, they must also add more codes. As a result, the company needs to create a standard operating procedure (that takes inputs from all relevant business teams) for updating the list of codes. Additionally, this procedure must ensure that the updated list of codes is properly inputted into Amazon’s data storage system.

With a robust list of codes, the company has two viable options for developing a method for accurate code selection: Standardize the work of the customer service agent or standardize the work of the customer.

Standardizing the work of the customer service agent is going to require either a revamp of the wizard software to remedy the slow performance issue, or development of an entirely new software program that will allow the CSA to perform the coding workflow simultaneously with the rest of their work. This software development requires massive coordination between many business teams and cannot be performed through the coordination of an intern and a single software developer. The operational excellence team that I worked with, and was developed to address work standardization issues, is the most appropriate coordinating device for these efforts.

Standardizing the work of the customer is going to require software development to be done on Amazon’s help pages. Further work needs to be accomplished in both refining the menu choices on the help pages, and ensuring that they are organized and worded in the most efficient manner, to allow the customer to correctly identify their issue. This development work must be done with input from the customer through the use of surveys, interviews, usability tests, etc.
References

Amazon.com Inc. (2008). *2007 annual report*


