

CASE STUDY

Written Use Cases Help Capture Tasks



Beth Agnew, Technical Communication Consultant, Agnew Communications

We all know that task-oriented documentation helps users more effectively than a manual that simply describes the product's functionality. Getting the time and resources to do a comprehensive task analysis, however, is an ongoing challenge for technical communicators.

We increasingly have to do more with less. To improve the bottom line, software companies rely on object-oriented CASE tools (Computer Aided Software Engineering) such as Rational Rose, and the Unified Modeling Language (UML) to develop new products on an iterative model. While that may help the programmers, it leaves the rest of the team at a disadvantage.

The drawback for non-technical staff is that information about the product's design is held internally, within the tool, for a longer period of time than if more traditional paper-based design methodologies were used. Where once we might have had access to a series of documents that covered different aspects of the design such as the requirements, specifications, and detailed design documents, we now have only relationship diagrams and models within the development application. Even when printed and distributed to those who do not have the development software, these technical diagrams are difficult for many to understand.

It's rather like trying to discern the characteristics of a baby in the womb, based on the sonogram. While miraculous developmental strides are taking place inside, those who are not actually involved aren't able to interpret the images; they can only wait and guess.

Proposed Best Practice Solution—Use Cases

Within the UML approach, however, is the concept of a use case—a scenario or description of how a particular feature is used by the user or “actor.” By adapting these use cases to a plain language narrative form, my teams have been able to get a jump-start on documentation and assist in achieving a collective vision of the product earlier in the software-development life cycle.

At one software-development company, my documentation team was faced with having to make sense of the sparse and highly technical use cases embedded within Rational Rose. We were about to begin writing the user guides for a complex, enterprise-wide information-management product, but we were overwhelmed by the internal technical details. Our users didn't care about multi-threading or database function calls; they needed to know how to get their documents processed by the system. And we knew we couldn't wait until the product was delivered to begin writing about it.

At the 1997 Toronto STC Conference, the workshop on Use Cases by Mary Nurminen and Leena M. Rasinaho of Nokia Telecommunications/Network Management Systems explained how to use narrative use cases to capture documentation requirements and the overall information requirements of end-users. Their presentation gave us a workable means of adapting the use case to our pressing needs.

Use cases defined and explained

Use cases are scenarios or step-by-step descriptions of typical user interactions with the product. Written in simple language, like a 10-year-old's essay, use cases state what happens and what happens next with no embellishment. This simple description of the user's interaction with the product comprises one task from beginning to completion. It explains that “The user will do the following with the product...” (see the table on page 80).

Business analysts should write the use cases, but if they won't or can't, then it is worthwhile for technical communicators to do so, even though it requires some additional research. Relying on the use case diagrams provided by developers within the CASE tool is not sufficient. There is too little detail, and many user tasks are missing. Narrative, external use cases, created by someone in the role of user advocate, focus the lens of the collective vision on the many connected details that make up user tasks. We found that the time spent in developing accurate use cases was an

Beth Agnew
Technical Communication
Consultant
Agnew Communications

bagnew@agnewcom.com

Beth Agnew is a technical communication consultant specializing in knowledge management and marketing communications. Currently working with Xenos Group, Inc. on their data to e-content product documentation strategies, Beth has previously worked with multimedia companies and other software-development groups. She has a BFA in Creative Writing from the University of Victoria and has been a popular speaker at STC conferences on the subject of writing for magazines. She welcomes your comments at <www.agnewcom.com>.

Parts of a Use Case

<u>Item</u>	<u>Notes</u>
<u>Title</u>	<ul style="list-style-type: none"> ◆ Operation covered in the use case ◆ Name of the “play” ◆ Not too broad
<u>Prepared by</u>	<ul style="list-style-type: none"> ◆ Name of the person writing the use case
<u>Version</u>	<ul style="list-style-type: none"> ◆ Reviewing and fine-tuning needed = versions ◆ Start with 1.0
<u>Purpose</u>	<ul style="list-style-type: none"> ◆ The actor wants to do this because... ◆ Gives reader quick understanding of use case
<u>Frequency</u>	<ul style="list-style-type: none"> ◆ How often operation is performed by actor ◆ If you don't have any idea, put “unknown” ◆ If you have a guess, put it with a (?) beside it
<u>Usability Requirements</u>	<ul style="list-style-type: none"> ◆ Describe acceptable performance boundaries for each task ◆ “Use case must be performed within x seconds,” or “Must allow unlimited repetitions,” and so on
<u>Actor(s)</u>	<ul style="list-style-type: none"> ◆ People who participate in carrying out operation ◆ For example: beginner, expert, or document specialist ◆ If your actor is a part of the program and not a person, you are describing an internal process
<u>Preconditions</u>	<ul style="list-style-type: none"> ◆ What has to be ready and available before starting the task ◆ Can be another use case
<u>Description</u>	<ul style="list-style-type: none"> ◆ What happens ◆ The story or play (group of scenarios) ◆ One single, recommended path of action ◆ Concrete, simple, written like a 10-year-old's essay ◆ Actor does this, then this, this happens, then this
<u>Exceptions</u>	<ul style="list-style-type: none"> ◆ What could go wrong ◆ Most important exceptions ◆ User's own mistakes not covered here
<u>Postconditions</u>	<ul style="list-style-type: none"> ◆ Final result of operation ◆ Goal achieved
<u>Menu Items (Optional)</u>	<ul style="list-style-type: none"> ◆ The menu items that are selected during this use case, if the product has an interface
<u>Index Entries</u>	<ul style="list-style-type: none"> ◆ Words related to this use case that could or should appear in the manuals and help indexes ◆ Each person working on the use cases should add his or her word choices ◆ Add any word you feel is appropriate for that topic ◆ Technical writer will edit this list

investment that provided a return later, when it came time to write the user guides.

Having a collection of tasks, embodied in approved use cases, meant that we could create draft manuals quickly. We could also compare

the use cases to earlier versions of the product, uncovering design elements that needed fixing. In one memorable exchange with the lead developer, we pointed out that the use case called for the user to print the document that

had been created on the system. Unfortunately, no print function had been included in the product. Because the product itself only prepared documents to be printed, the developers had not been asked to consider the printing operation. What they took for granted was a critical issue to our users. Fortunately, this flaw was identified early in the cycle, and the fix took very little development time and few resources.

In addition to task descriptions, use cases should also include statements about preconditions, postconditions, and exceptions (error conditions): what conditions existed before the task began, what conditions should exist when the task is successfully completed, and what can go wrong. A postcondition of “document is printed” in that use case saved the company from potential embarrassment and costly expenditure of resources. Perhaps the error would have been found during the QA cycle but at a greater cost to repair.

My team adapted the use case format to include lists of index entries so that they could assist in capturing the terminology for the indexes. Because different people think of tasks using different terminology, each person who reviews the use case adds their terms to the list. For applications with a complex graphical user interface, we also added a list of the menu items and icons that apply to the tasks described in the use case.

Use Cases are an Easy Sell

The great selling point within the company for our efforts was that use cases are beneficial across the spectrum of people involved in product development. Product marketing values them because they map to one or more product requirements. Developers and Quality Assurance can ensure the product meets the

specifications that have been fully described in the use case. The testing engineers can even begin to write test cases based on the use cases. Management, other non-technical personnel, and the marketing communications group can read the use cases to understand what the product will actually do for the customer. Customer support can begin to build a knowledge base of potential problems and solutions because the use cases include an optimal workflow, as well as error conditions and exceptions.

Writing Use Cases

Use cases should be very simple and straightforward, written in plain language. The sequence of actions must represent an actual workflow and represent the sequence of steps the user takes to perform a single task. The focus is on correctly describing each part of the task, without elaboration.


Each use case covers only *one* task or operation, although it may represent a sequence of activities. If you have different options within the sequences, then you need to write additional use cases for them. Generally, each use case requires about a page to write.

Use cases review

Once the use cases are written, they should be verified. We held a formal review in which all stakeholders in the product-development process had an opportunity to check the use cases for completeness and adherence to specifications and requirements. We distributed the complete group of use cases to the reviewers about a week in advance. The review team typically included a senior developer, a lead technical writer, a business/systems analyst, the product manager, and representatives from quality assurance, customer support, and the implementation team or professional services group because they were involved in working

Results Achieved with Use Cases

Use cases

- ◆ Prove the design
- ◆ Point out any “holes” in the workflow
- ◆ Externalize the design
- ◆ Promote common terminology
- ◆ Provide opportunity to build a collective vision
- ◆ Help communicate the design to non-technical groups such as marketing and sales
- ◆ Give technical writers a head start on the documentation
- ◆ Involve technical writers and development together early in the software-development life cycle
- ◆ Provide a framework for task analysis and a method for capturing user tasks 

with the product at customer sites. The more user-oriented focus you apply to the use cases, the more valuable they will be in capturing task information.

Use Cases Help in Design

If you can't come up with a use case for each of the features and functions in a product or application, then those functions may not be well thought out. You may even need to ask if there is a business need for the functionality if no use case can be drawn. Why spend precious development time and resources on something users don't need?

Similarly, if you find it difficult to connect the use cases to each other, a coherent workflow may be absent. Workflow diagrams help you connect the use cases during development.


When you identify the actor(s) in a use case, you may realize that there are multiple types of users and multiple audiences that you need to write for. If different user types per-

form different tasks, they should be the subject of different use cases.

Any design-based industry, including manufacturing, can benefit from use cases. The actor or user is the central character in the "play" or scenario. The actor might be the operator of the equipment, the user of the product, or even the recipient of the service. Try writing a series of use cases for customer support calls and see if they point out any weaknesses in your process design.

In conclusion, we have found narrative use cases to be valuable aids to capturing user tasks and assisting all members of the development team to understand the product from the customer's point of view.

Sample Use Case

The following use case covers complex actions that intersect with additional use cases. There is an overall task represented by this use case and subtasks covered by additional use cases. Note the exceptions and error conditions that are mentioned. 

Cooking a Barbecue

Title	Cooking a Barbecue
Version	1.0
Prepared By	Martha Stewart
Purpose	A cook wants to make a barbecue consisting of hamburgers, ribs, potato salad, and tossed salad to serve to her guests.
Frequency	Occasionally
Usability Requirements	It must be possible to prepare the barbecue so that all food is ready to be served to the guests at the right temperature at the same time.
Actor(s)	Cook
Preconditions	All ingredients, utensils, and a barbecue grill at hand
Description	Cook prepares potato salad. [Use Case: Preparing Potato Salad] Cook prepares tossed salad. [Use Case: Preparing Tossed Salad] Cook lights barbecue grill and cooks hamburgers. (Exception: Barbecue grill will not work) [Use Case: Cooking Hamburgers] Cook prepares ribs. [Use Case: Barbecuing Ribs] (Exception: Food is spoiled) Cook places food on plates to be served. (Exception: Unexpected guests) (Exception: No guests) [Use Case: Serving Food]
Exceptions	Barbecue grill will not work: occurs if fuel has been depleted or briquets do not light Food is spoiled: occurs if food has not been properly refrigerated beforehand Unexpected guests: occurs if more people than anticipated show up for the barbecue No guests: occurs if no one shows up for the barbecue
Postconditions	Food is ready to be served.
Index Entries	cooking; barbecue; barbecue, cooking; hamburgers; potato salad; ribs; tossed salad; entertaining; coordinating a meal

Sample Use Cases

1. Adding a Comment

Title:	Adding a Comment
Version:	1.0
Prepared by:	Jane Smith
Purpose:	A user wants to make a comment at a specific spot in a PDF format document.
Frequency:	Once a week (?); Multiple times during an editing session.
Usability Requirements:	User must be able to add a comment at any point in the PDF document.
Actor(s):	User
Preconditions:	Must have a version of Adobe Acrobat with annotation capability.
Description:	The user opens the PDF document in Acrobat and goes to the place where she wants to add a note. She marks the spot where he wants to note to be added and gives the command that begins the comment attachment process. (<i>Note: you do not have to know exactly what this command is for the use case to be written.</i>) Then the user writes the note. (Exception: Unable to save)
Exceptions:	Unable to save: occurs if the user does not have the proper privileges to attach notes to the PDF.
Postconditions:	A new comment is placed on the PDF document.
Menu Items:	Tools > Notes;
Index Entries:	Notes, Adding; Add - Note; Adding a Note; Annotations; Additional Information; Comments; User Notes, Editing Notes.

2. Changing a Flat Tire

Title:	Changing a Flat Tire
Version:	1.2
Prepared by:	Otto M. Obile
Purpose:	Driver wants to change a flat tire so that he can drive the car.
Frequency:	Infrequently
Usability Requirements:	Driver must be able to change the tire within 20 minutes.
Actor(s):	Driver
Preconditions:	There must be a flat tire on an otherwise serviceable car. There must be a spare tire available. [Use Case: Maintaining the spare tire] There must be a jack, lug nut wrench (tire iron), wheel chock, and lubricant available.
Description:	Driver puts the car in park in a level area and applies the parking brake. (Exception: Parking brake unserviceable.) He turns off engine, turns on hazard lights, and opens hood to indicate that he is stopped for repairs. He places a wheel chock or large rock behind or in front of the diagonally opposing wheel to prevent the car from rolling. He removes hubcap if necessary. He loosens the lug nuts before jacking up the car. [Use Case: Loosening lug nuts] He jacks up the car safely. [Use Case: Jacking up a car] (Exception: No jack available.) He removes the lug nuts [Use Case: Removing lug nuts] He removes the flat tire. He mounts the spare tire. (Exception: No spare) (Exception: Spare is Flat) He replaces the lug nuts. He lowers the jack. He tightens the lug nuts. He replaces the hubcap if necessary.
Exceptions:	Parking Brake Unserviceable: occurs if the parking brake has not been properly maintained. No jack: occurs if the car is not equipped with a jack. No spare: occurs if no spare tire is available Spare is flat: occurs if the spare tire has not been inflated.
Postconditions:	Tire is changed and car is ready to be driven. [Use Case: Driving a car]
Index Entries:	Tires, flat; Changing a Tire; Flat tire; Spare tire; Equipment: Jack; Equipment: Lug Nut Wrench; Roadside maintenance;

3. *Cooking a Brunch*

This use case covers complex actions that intersect with additional use cases. There is an overall task represented by this use case, and subtasks covered by additional use cases. Note the exceptions and error conditions that are mentioned.

Title:	Cooking a Barbecue
Version:	1.0
Prepared by:	Martha Stewart
Purpose:	A cook wants to make a barbecue consisting of hamburgers, ribs, potato salad, and tossed salad, to serve to her guests.
Frequency:	Occasionally
Usability Requirements:	It must be possible to prepare the brunch so that all food is ready to be served to the guests at the right temperature at the same time.
Actor(s):	Cook
Preconditions:	All ingredients, utensils, and a barbecue grill at hand.
Description:	Cook prepares potato salad. [Use Case: Preparing Potato Salad.] Cook prepares tossed salad. [Use Case: Preparing Tossed Salad] Cook lights barbecue grill and cooks hamburgers. (Exception: Barbecue grill will not work.) [Use Case: Cooking Hamburgers] Cook prepares ribs. [Use Case: Barbecuing Ribs] (Exception: Food is spoiled.) Cook places food on plates to be served. (Exception: Unexpected guests.) (Exception: No guests.) [Use Case: Serving Food.]
Exceptions:	<ul style="list-style-type: none">• Barbecue grill will not work: occurs if fuel has been depleted or bricquets do not light.• Food is spoiled: occurs if food has not been properly refrigerated beforehand.• Unexpected guests: occurs if more people than anticipated show up for the barbecue.• No guests: occurs if no one shows up for the barbecue.
Postconditions:	Food is ready to be served.
Index Entries:	Cooking - Barbecue; Barbecue, Cooking; Hamburgers; Potato Salad, Ribs, Tossed Salad; Entertaining; Coordinating a Meal.

Sample Use Cases – problems encountered during writing

This sample indicates what might happen as you are creating a use case, and how to deal with it.

4. Formatting Text (in MS Word)

Title:	Formatting Text
Version:	3.2
Prepared by:	Beth Agnew
Purpose:	A user wants to apply <i>various formatting options</i> to text in a document. <As soon as you write language that indicates multiple options, you should realize that you may need multiple use cases.>
Frequency:	Multiple times during an editing session; many times an hour.
Usability Requirements:	User must be able to format any of the text in the document, including Header and Footer text. <You're going to need use cases for Headers and Footers as well.>
Actor(s):	User
Preconditions:	Must have a document open in the editing window.
Description:	The user highlights the text to be formatted. [Use Case: Highlighting Text]. User selects the type of formatting to be applied. <This is not specific enough. It is a high level task description, and use cases break down to lower level tasks. You should have use cases for changing fonts, applying styles such as Bold, Italic, etc., and for formatting bullets, numbers, columns, and so on. However, these may be sub-cases under this particular use case to show the relationship between them.> (Exception: Unable to format text)
Exceptions:	Unable to format text: occurs if the user... <What error conditions might keep the user from being able to format the text? Are there any? If not, there's no exception. Or, the exception may be related to something else, such as not being able to highlight the text.>
Postconditions:	Text is formatted as intended.
Menu Items:	Format > Font; Format > Paragraph; Format > Bullets; <Too many menu items indicate that you need multiple use cases.>
Index Entries:	Text, Formatting; Bold text, Italic text, Fonts, changing; Changing Fonts; Formatting, text