



# The Ambiguity Review Process

© Richard Bender  
Bender RBT Inc.  
17 Cardinale Lane  
Queensbury, NY 12804  
518-743-8755  
[rbender@BenderRBT.com](mailto:rbender@BenderRBT.com)

## **The Ambiguity Review Process**

### **Purpose:**

An Ambiguity Review improves the quality of requirements by making them deterministic, unambiguous, correct and complete. An Ambiguity Review is a testing technique that helps eliminate defects in the requirements phase of the software development lifecycle, thereby preventing those defects from propagating to the remaining phases of the software development lifecycle.

### **Description:**

The Ambiguity Review process is a two step process. The initial Ambiguity Review is performed by someone who is not a domain expert, and is not reading the requirements for content, but only to identify ambiguities in the logic and structure of the wording. The Ambiguity Review takes place after the requirements (or a section of the requirements) reach first draft. This review finds all of the generic ambiguities such as unclear references. Since the initial reviewer is not a domain expert they cannot read into the specification facts that are not explicitly there.

Once the issues identified in the initial review have been addressed, the requirement is then reviewed for content (i.e., correctness and completeness) by domain experts.

### **Deliverables:**

The Ambiguity Review deliverables include the following:

- If the requirements are in a document, then the ambiguities are documented on either a copy of the requirements or in a separate document.
- If the requirements are stored in a requirements management tool such as CaliberRM, then the ambiguities are documented directly in tool.
- A summary of the Ambiguity Review findings.
- Optionally, if a defect tracking tool is being used, all defects found in the initial Ambiguity Review are logged as one incident, with the number of issues noted. After the next revision of the requirements, if open issues remain, they are logged as individual incidents.

## The Ambiguity Review Checklist:

The Ambiguity Review Checklist powers the Ambiguity Review Process. The Ambiguity Review Checklist identifies 15 common problems that occur in writing requirements.

- ❖ Dangling else
- ❖ Ambiguity of reference
- ❖ Scope of action
- ❖ Omissions
  - ◆ Causes without effects
  - ◆ Missing effects
  - ◆ Effects without causes
  - ◆ Complete omissions
  - ◆ Missing causes
- ❖ Ambiguous logical operators
  - ◆ Or, And, Nor, Nand
  - ◆ Implicit connectors
  - ◆ Compound operators
- ❖ Negation
  - ◆ Scope of negation
  - ◆ Unnecessary negation
  - ◆ Double negation
- ❖ Ambiguous statements
  - ◆ Verbs, adverbs, adjectives
  - ◆ Variables, unnecessary aliases
- ❖ Random organization
  - ◆ Mixed causes and effects
  - ◆ Random case sequence
- ❖ Built-in assumptions
  - ◆ Functional/environmental knowledge
- ❖ Ambiguous precedence relationships
- ❖ Implicit cases
- ❖ Etc.
- ❖ I.E. versus E.G.
- ❖ Temporal ambiguity
- ❖ Boundary ambiguity

As an example, one of the Ambiguity Review Checklist items is the Dangling Else. A Dangling Else can be identified when one of the following sets of words is used in a sentence:

**MUST BE, WILL BE, IS ONE OF, SHOULD BE, COULD BE, CAN BE, or SHALL.**

As an example, an excerpt from a set of requirements states the following:

“The Marriage Status must be either Married, Single, or Divorced.”

This requirement states what happens under normal circumstances, or the “go” path. However, it is not a complete requirement, because it does not describe what happens if we are off the “go” path. What is the exception or error condition if another value is entered in the Marriage Status, such as Separated?

## List of Words that Point to Potential Ambiguities

Many ambiguities referred to in the Ambiguity Review Checklist items can be identified by looking for key words and phrases in the requirements. The following list of words point to potential Ambiguities:

### Dangling Else

can	could	is one of
must	shall	should
will	would	

### Ambiguity of Reference

above	below	it
such	the previous	them
these	they	this
those		

### Ambiguous Adjectives

all	any	appropriate
custom	efficient	every
few	frequent	improved
infrequent	intuitive	invalid
many	most	normal
ordinary	rare	same
seamless	several	similar
some	standard	the complete
the entire	transparent	typical
usual	valid	

### Ambiguous Adverbs

accordingly	almost	approximately
by and large	commonly	customarily
efficiently	frequently	generally
hardly ever	in general	infrequently
intuitively	just about	more often than not
more or less	mostly	nearly
normally	not quite	often
on the odd occasion	ordinarily	rarely
roughly	seamlessly	seldom
similarly	sometime	somewhat
transparently	typically	usually
virtually		

### **Ambiguous Variables**

the application  
the database  
the frame  
the module  
the screen  
the table

the component  
the field  
the information  
the page  
the status  
the value

the data  
the file  
the message  
the rule  
the system  
the window

### **Ambiguous Verbs**

adjust  
calculate  
compute  
customize  
edit  
indicate  
maximize  
might  
perform  
provide  
validate

alter  
change  
convert  
derive  
enable  
manipulate  
may  
modify  
process  
support  
verify

amend  
compare  
create  
determine  
improve  
match  
minimize  
optimize  
produce  
update

### **E.G. versus I.E.**

e.g.

i.e.

### **Implicit Cases**

also  
besides  
for all other  
in addition to  
notwithstanding  
still  
whereas  
as necessary

although  
but  
furthermore  
likewise  
on the other hand  
though  
yet

as well  
even though  
however  
moreover  
otherwise  
unless  
as required

### **Temporal Ambiguity**

after  
at the appropriate time  
daily  
fast  
monthly  
soon  
weekly

annually  
bimonthly  
every other month  
in a while  
quarterly  
twice a month  
yearly

at a given time  
biweekly  
every other week  
later  
quickly  
twice a year

### **Boundary Ambiguity**

up to

among

including

### **Totally Ambiguous**

etc.

sentences that end with ?

## **Ambiguity Review Metrics**

In a typical Ambiguity Review, 15 pages of requirements can be reviewed for Ambiguity and documented per day. Using a requirements management tool, such as CaliberRM, the equivalent of 25 pages of requirements can be reviewed for ambiguity and documented per day. The increased efficiency occurs because of the improved logistics in making the requirement available to the reviewers and the increased visibility of the comments from the various reviewers without the need to manually merge them and redistribute them. They are entered into the data base with the requirement itself.

In the initial reviews of the work product of a given author it is common to find about ten or so ambiguities per page of detailed requirements. After a few requirements from that same author have been reviewed the number of ambiguities found tends to drop by a factor of 20X. What is happening is that once the author realizes that people are looking for such things as “dangling elses”, then the next requirement does not contain any such problems. The process leads quickly to defect avoidance.

The cost per ambiguity found is only a few dollars. Experience has shown that if these ambiguities are not addressed prior to design and coding, nearly 100% of them will result in defects in the code. Not finding them until integration testing or later results in costs hundreds of times more expensive per defect.

### **Benefits of an Ambiguity Review:**

- Higher quality requirements are made available to the domain experts to read for correctness and completeness.
- Defects are corrected at the earliest point in the software development lifecycle (defect avoidance instead of defect identification in latter phases of the software development lifecycle).
- The cost of correcting defects is at its lowest point in the software development lifecycle.
- Timely feedback from the Ambiguity Review reduces issue resolution time.
- All members of the Project Team can work from one clear set of requirements, thereby reducing the chance of scrap and rework throughout the software development lifecycle.

## **Ambiguity Review Training**

The Ambiguity Review Process is taught in three courses offered by Bender RBT Inc. These courses are:

- Finding Ambiguities in Requirements (one day), aimed at anyone who has to read requirements
- Writing Testable Requirements (three days), aimed at anyone who has to write requirements
- Requirements-Based Testing (three days), aimed at anyone who has to test software