The Ambiguity Review Process

© Richard Bender
Bender RBT Inc.
17 Cardinale Lane
Queensbury, NY 12804
518-743-8755
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

<table>
<thead>
<tr>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>can</td>
<td>could</td>
<td>is one of</td>
</tr>
<tr>
<td>must</td>
<td>shall</td>
<td>should</td>
</tr>
<tr>
<td>will</td>
<td>would</td>
<td></td>
</tr>
</tbody>
</table>

#### Ambiguity of Reference

<table>
<thead>
<tr>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>above</td>
<td>below</td>
<td>it</td>
</tr>
<tr>
<td>such</td>
<td>the previous</td>
<td>them</td>
</tr>
<tr>
<td>these</td>
<td>they</td>
<td>this</td>
</tr>
<tr>
<td>those</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Ambiguous Adjectives

<table>
<thead>
<tr>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>any</td>
<td>appropriate</td>
</tr>
<tr>
<td>custom</td>
<td>efficient</td>
<td>every</td>
</tr>
<tr>
<td>few</td>
<td>frequent</td>
<td>improved</td>
</tr>
<tr>
<td>infrequent</td>
<td>intuitive</td>
<td>invalid</td>
</tr>
<tr>
<td>many</td>
<td>most</td>
<td>normal</td>
</tr>
<tr>
<td>ordinary</td>
<td>rare</td>
<td>same</td>
</tr>
<tr>
<td>seamless</td>
<td>several</td>
<td>similar</td>
</tr>
<tr>
<td>some</td>
<td>standard</td>
<td>the complete</td>
</tr>
<tr>
<td>the entire</td>
<td>transparent</td>
<td>typical</td>
</tr>
<tr>
<td>usual</td>
<td>transparent</td>
<td>valid</td>
</tr>
</tbody>
</table>

#### Ambiguous Adverbs

<table>
<thead>
<tr>
<th>Word</th>
<th>Word</th>
<th>Word</th>
</tr>
</thead>
<tbody>
<tr>
<td>accordingly</td>
<td>almost</td>
<td>approximately</td>
</tr>
<tr>
<td>by and large</td>
<td>commonly</td>
<td>customarily</td>
</tr>
<tr>
<td>efficiently</td>
<td>frequently</td>
<td>generally</td>
</tr>
<tr>
<td>hardly ever</td>
<td>in general</td>
<td>infrequently</td>
</tr>
<tr>
<td>intuitively</td>
<td>just about</td>
<td>more often than not</td>
</tr>
<tr>
<td>more or less</td>
<td>mostly</td>
<td>nearly</td>
</tr>
<tr>
<td>normally</td>
<td>not quite</td>
<td>often</td>
</tr>
<tr>
<td>on the odd occasion</td>
<td>ordinarily</td>
<td>rarely</td>
</tr>
<tr>
<td>roughly</td>
<td>seamlessly</td>
<td>seldom</td>
</tr>
<tr>
<td>similarly</td>
<td>sometime</td>
<td>somewhat</td>
</tr>
<tr>
<td>transparently</td>
<td>typically</td>
<td>usually</td>
</tr>
<tr>
<td>virtually</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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, alter, amend
calculate, change, compare
compute, convert, create
customize, derive, determine
edit, enable, improve
indicate, manipulate, match
maximize, may, minimize
might, modify, optimize
perform, process, produce
provide, support, update
validate, verify

E.G. versus I.E.

e.g. i.e.

Implicit Cases

also, although, as well
besides, but, even though
for all other, furthermore, however
in addition to, likewise, moreover
notwithstanding, on the other hand, otherwise
still, though, unless
whereas, yet, as required
as necessary

Temporal Ambiguity

after, annually, at a given time
at the appropriate time, bimonthly, biweekly
daily, every other month, every other week
fast, in a while, later
monthly, quarterly, quickly
soon, twice a month, twice a year
weekly, yearly

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 database 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