AIR UNIVERSITY multan campus

Software Engineering

Muhammad Uzair

153191

**Ambiguous Requirements:**

Many software requirements suffer from ambiguity. Ambiguity means that a single reader can interpret the requirement in more than one way or that multiple readers come to different interpretations. In either case, [ambiguous requirements lead to confusion](http://searchsoftwarequality.techtarget.com/feature/Writing-requirements-Common-sense-measures-for-success), wasted effort and rework.

Following are the examples of Ambiguous Requirements:

## **Syntactic Ambiguity:**

[Syntactic ambiguity](http://en.wikipedia.org/wiki/Syntactic_ambiguity) occurs when a sequence of words can be given more than one grammatical structure. For example, consider the sentence:

*The police shot the rioters with guns*

This can be interpreted as the rioters with guns were shot by the police, or that the rioters were shot by police with guns; the grammatical structure of the sentence is ambiguous.

**Pronouns:**  
Pronouns offer another opportunity for confusion if the antecedent is for each pronoun is not absolutely clear. If you say "this" or "that," there should be no confusion in the reader's mind as to what you are referring to.

# **Writing Unambiguously:**

A key part of writing unambiguous sentences is to be able to recognize an ambiguous phrase as you write it. In the section on [detecting ambiguity](https://shane.io/2013/04/22/ambiguity-in-software-specs.html#detecting_ambiguity), three core types of ambiguity were identified: lexical, syntactic, and semantic. We’re going to go over some common examples, identifying the ambiguity in each and fixing them.

# **Detecting Ambiguity:**

When ambiguity exists in writing, there are different possible underlying causes of the ambiguity. These kinds of ambiguity are outlined below. Note that in the academic literature there are a few more than are mentioned here, but they start to blur the distinction between the different forms, and so they have been omitted in the interest of keeping the conceptual differences clear.

**References**

1. [**http://searchsoftwarequality.techtarget.com/tip/Ambiguous-software-requirements-lead-to-confusion-extra-work**](http://searchsoftwarequality.techtarget.com/tip/Ambiguous-software-requirements-lead-to-confusion-extra-work)
2. [**https://shane.io/2013/04/22/ambiguity-in-software-specs.html**](https://shane.io/2013/04/22/ambiguity-in-software-specs.html)
3. [**https://link.springer.com/chapter/10.1007/3-540-28244-0\_11**](https://link.springer.com/chapter/10.1007/3-540-28244-0_11)
4. [**https://www.tutorialspoint.com/software\_engineering/software\_requirements.htm**](https://www.tutorialspoint.com/software_engineering/software_requirements.htm)