Air University (Multan Campus)

Dept. Computer Sciences, BSCS-V Fall'2017

Ambiguous Requirements

Subject: Software Engineering

Submitted to:

Sir Ahmad Mohsin

Submitted by:

Umamah Ayyaz Ansari (153174)

Date of Submission:

5th October, 2017

Ambiguous Requirements

The origin of Ambiguous is from “Ambiguity” which means something that does not have single clear meaning. Ambiguous means able to be understood in more than one way. It depends on the reader’s perception.

Ambiguous requirements can lead us to many problems because user stories can be interpreted in many different ways. This can cause miscommunication and eventually lead to software or product been built incorrectly. This wastes resources like time and money.

The user stories is written in natural language and can lead us to ambiguity because it will give the team members new requirements every time they read them.

Examples

1. Use of Indirect works like “Should” , “could” , “will” or “may” can lead to ambiguous requirements:

“As an online poker player, I may win the money so that I am rewarded for winning game.”

In this statement, it is unclear as to the circumstances in which the online poker player wins money. You have to clarify this user story by specifying that information.

It should be written as,

“As an online poker player, I receive money when I win a game of poker against my opponents, so that I am rewarded for a winning game.”

2. “As an Admin, I want every social media users to use an account so that I can track their usage.”

In this example, we are not sure which account they are referring too. Is it One account accessed by everyone or they all have their own account.

3. Propositional word problem:

   “Entry A should appears before Entry B in the database”

   According to the statement it is not confirm that Entry B should be entered consecutive to Entry A or there are other entries to in between.

   If we took the case of consecutive entries then the user story should be written as,

   “Entry B appears in the database followed by Entry A” OR

   “Entry B must appear right after Entry A”.

Conclusion:

Writing unambiguous requirements requires us to write complete requirements. It also requires us to use precise language that communicates information across domains to our readers. To determine the right level of effort, we need to monitor the effectiveness of our communication, and balance that with the amount of time we can afford to dedicate to word understanding instead of other product management activities.

References:

- http://tynerblain.com/blog/2006/06/12/writing-unambiguous-requirements/
- http://searchsoftwarequality.techtarget.com/tip/Ambiguous-software-requirements-lead-to-confusion-extra-work
- https://shane.io/2013/04/22/ambiguity-in-software-specs.html
- https://cs.uwaterloo.ca/~dberry/ambiguity.res.html
-