Miggy Reyes

Professor Burgess

ENGR-2392-H01

29 September 2018

State Position: ACM Theory. Use the Pluralistic-Act-Utilitarianism and RFP mixed together. You want the best experience for you consumers. But you also do not want to get stagnant in your learning. You should always be “cultivating and practicing your essential capacities”. Make sure that the work that you do is impactful.

Prompts:

* Do you think one theory should be used as the primary basis of your discipline’s Code? Exclusive basis? If so, which one and why? If not, why not?
  + Pluralistic-Act-Utilitarianism mixed with a little bit of Virtue Ethics
  + Virtue Ethics because engineers should always be innovating and using their education to the fullest of their capabilities
  + Utilitarianism because there should be more to life than just pleasure for software engineers and computer scientists.
* Identify and address the strongest counterargument to your position
  + If we only cared about what was best for the most people, we would ignore people with disabilities since they are the minority. We should be creating software with accessibility.

Structure:

1. Intro
   1. Basis of what the Code of Ethics is about for CS
   2. It uses Utilitarianist theory and Virtue Ethics
   3. However, we should still consider the minority so that software is accessible to everyone
2. Why we should be using Utilitarianist theory
   1. Pluralistic
      1. There is more to life than just pleasure
      2. People need other people and through friendship, a possible intrinsic good, people are ethical when the create these types of relationship
   2. Act
      1. People are too diverse, and it does not make sense to categorize every single interaction between 2 different people
      2. Enjoying things is nothing to be ashamed of, but I agree that too much of one thing is a bad thing. (Use an example)
3. Why we should be using Virtue Ethics theory
   1. Virtue Ethics because engineers should always be innovating and using their education to the fullest of their capabilities
   2. Honesty about software is key. We should be clear with what we are building and what our software should be used for.
   3. Abuse of software should be kept in mind when creating software and actions should be taken towards it
   4. There are times where the only way to solve a problem is with software. Following practical wisdom is very important.
4. What is bad about
   1. Utilitarianism is all about consequences, it doesn’t care about the minority as much. In CS we will need to consider the minority.
   2. If we only cared about what was best for the most people, we would ignore people with disabilities since they are the minority. We should be creating software with accessibility.
   3. Need to make software for low-performance phones and machines