Denise Hendricks

CSET 115

10/18/21

Discussion & Debate Lecture 1

1. Provide several examples (both positive and negative) that indicate the impact of software on our society
   1. Writing a faulty software could unreversible damage like a faulty medical machine could lead to harm of the patient. (Negative)
   2. Software wouldn’t make human error so things like surgical procedures would be going well rather than bad. (Positive)
2. Many modern applications change frequently—before they are presented to the end user and then after the first version has been put into use. Suggest a few ways to build software to stop deterioration due to change.
   1. By allowing automatic updates to keep the software “up to date”
   2. The software should be made in a way that changes should be quite easy to change.
3. As software becomes more pervasive, risks to the public (due to faulty programs) become an increasingly significant concern. Develop a doomsday but realistic scenario in which the failure of a computer program could do great harm (either economic or human).
   1. If there was a faulty program in a life support system, it could cause a patients life support to fail causing death of the patient
4. Add two additional myths to the list presented in this lecture. Also state the reality that accompanies the myth.
   1. Adding more people to a project would make things go faster
      1. Adding more people means someone must explain to the newcomers how the project is going to work therefore delaying the project itself
   2. Adding Features is a piece of cake
      1. Proper documentation ensure that the product would be good quality. And you cannot create proper document without clear list of requirements.
5. Answer the following questions -
   1. Why does it take so long to get software finished?
      1. Because when presented to the customer for the first time the customer could revise the software in massive ways and not realize how big of an ordeal it would be.
   2. Why are development costs so high?
      1. Due to the amount of time and people it takes to develop a program
   3. Why can’t we find all errors before we give the software to our customers?
      1. Because if it is a complex software, it wouldn’t possibly be able to predict every execution path
   4. Why do we spend so much time and effort maintaining existing programs?
      1. Because when changes and updates are made it could deteriorate the existing program.
   5. Why do we continue to have difficulty in measuring progress as software is being developed and maintained?
      1. Because each task breaks down into numerous smaller pieces, because computers must be told every little thing. Therefore, you cannot measure progress when you have so many tasks to keep track of.