1.1. Provide at least five additional examples of how the law of unintended consequences applies to computer software.

Internet dating: Mobile applications change the way of dating

Communication: Visual communication through software

Intelligence: Acquiring Intel on different crimes thus enhancing national security

Health sector: A diagnosis of diseases.

Education: Accessing e-learning materials like books and libraries.

1.2. Provide a number of examples (both positive and negative) that indicate the impact of software on our society.

Positive

- ✓ Get things done easily
- ✓ Ability to do research without going to a library
- ✓ Online marketing
- ✓ Eases school registration in larger institutions.
- ✓ Improved communication for example the use of cell phones.
- ✓ Health improvement when accessing patients records
- ✓ Efficiency in the work fields

Negative

- ✓ Online gambling
- ✓ Inability to make personal interactions
- ✓ Unauthorized access of public data
- ✓ Unemployment
- ✓ Addiction

1.3. Develop your own answers to the five questions asked at the beginning of Section 1.1. Discuss them with your fellow students.

• Why does it take so long to get software finished?

Poor planning

Lack of resources

Complexity

• Why are development costs so high?

Size of software

Third party dependences

Change of customer requirements

AGUMYA NYSON
2023BSE005PS

Complexity of the software.

• **Why can’t we find all errors before we give the software to our customers?**

Changing preference

Human Error

Incomplete testing

• **Why do we spend so much time and effort maintaining existing programs?**

To improve performance

Adaption to changing environments

Improve satisfaction

• **Why do we continue to have difficulty in measuring progress as software is being developed and maintained?**

Changing in customer requirements

There’s no standard way of measuring progress due to the different programming languages used to develop the programs.

Incomplete testing

1.3. 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.

Software applications should be maintainable. This implies that they should be engineered to a degree that changes can be made rather easily as the application grows. One way to minimize deterioration due to change is to allow automatic updates to be built in. Take an example of Windows OS, it has the option to allow automatic updates for necessary security and firewall Platforms to ensure that system is always up to date. Since previous applications are always being updates, it is important to build new software with the same capabilities.

1.5. Consider the seven software categories presented in Section 1.1.2. Do you think that the same approach to software engineering can be applied for each? Explain your answer.

No, Software Engineers worldwide are hard at work on software projects in one or more of these categories. In some cases, new systems are built, but in many others, existing applications are being corrected, adapted, and enhanced. Because of this, a different approach to software engineering may be required for individual categories .Many of the programs that software engineers work on are extremely old, and continue to be upgraded. Therefore, it makes sense that you wouldn't use the same approach for an existing program that you would use for a new program.

AGUMYA NYSON
2023BSE005PS