Shashank Thakare

Assignment 16 July 2019

Q1. What is Software Maintenance

A. The life of your software does not end when it finally launches. In reality, its life has just begun.

Software is always evolving and it is never finished as long as it is used partly to accommodate for the ever changing world we live in. The evolution of your software might be motivated by a variety of reasons; to keep the software up and running, to upgrade to the latest release, enhance features or to rework the system for future maintainability. No matter the motivation, software change is vital for the evolution and success of it. Therefore, software will have to undergo changes, and understanding the different types of changes your software can go through is important to realize that software maintenance is more than just bug fixing.

There are four categories of software change:

* Corrective
* Adaptive
* Perfective
* Preventive

### Corrective Change

Corrective change, most commonly referred to as “bugs,” is the most typical change associated with maintenance work. Corrective changes address errors and faults in your software that could affect various areas of your software; design, logic or code.

### Adaptive Change

Adaptive change is triggered by changes in the environment your software lives in. An adaptive change can be triggered by changes to the operating system, hardware, software dependencies and even organizational business rules and policies.

### Perfective Change

Perfective changes refers to the evolution of requirements and features in your existing system. As your software gets exposed to users they will think of different ways to expand the system or suggest new features that they would like to see as part of the software, which in turn can become future enhancements to the system.

### Preventive Change

Preventive changes refer to changes made to increase the understanding and maintainability of your software in the long run. Preventive changes are focused in decreasing the deterioration of your software in the long run. Restructuring, optimizing code and updating documentation are common preventive changes.