SUBJECT: SOFTWARE TESTING AND QUALITY ASSURANCE
TOPIC : COMPATIBILITY TESTING
NAME OF TEACHER: SIMNA V J
ACADEMIC YEAR: 2020-2021

COMPATIBILITY TESTING

COMPATIBILITY TESTING

- Testing done to ensure that the product features work consistently with different infrastructure components is called compatibility testing

- Infrastructure parameters could be of hardware,software or other components

- **Parameters** that generally affect compatibility of product are
- Processor
- Resource availability on machine
- Operating system
- Architecture and characteristics of machine etc

- In order to arrive at practical combinations of the parameters to be tested, a compatibility matrix is created

- A compatibility matrix has as it's columns various parameters the combinations of which have to be tested.

- Each row represents a unique combination of a specific set of values of the parameters

- Common techniques used for performing compatibility testing using compatibility table are

  1. Horizontal combination:

     All values of parameters that can coexist with the product for executing the set cases are grouped together as a row in the compatibility matrix

2. Intelligent sampling

- Combination of infrastructure parameters are combined with the set of features intelligently and tested

- The selection of intelligent samples is based on information collected on the set of dependencies of the product with the parameters.

- If the product results are less dependent on a set of parameters then they are removed from the list of intelligent samples

- Compatibility testing not only includes parameters that are outside the product, but also includes some parameters that are a part of product

- Compatibility testing of a product involving parts of itself can be divided into two

1.BACKWARD COMPATIBILITY

- The testing that ensures the current version of the product continues to work with the older versions of the same product is called backward compatibility testing

2. FORWARD COMPATIBILITY TESTING

- There are some provisions for the product to work with later versions of the product and other infrastructure components, keeping future components in mind