**Software Engineering Assignment**

**Rework**

Rework in software development is the additional effort of redoing a process or activity that was incorrectly implemented in the first instance or due to changes in requirements from clients.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Verification** is the process of checking that a software achieves its goal without any bugs.

Verification means **Are we building the product, right?**

**Validation** is the process of checking whether the software product is up to the mark or in other words product has high-level requirements.

Validation means **Are we building the right product?**

**Difference between Verification and Validation**

|  |  |
| --- | --- |
| **Verification** | **Validation** |
| It includes checking documents, designs, codes, and programs. | It includes testing and validating the actual product. |
| Verification is static testing. | Validation is dynamic testing. |
| It does *not* include the execution of the code. | It includes the execution of the code. |
| Methods used in verification are reviews, walkthroughs, inspections, and desk-checking. | Methods used in validation are Black Box Testing, White Box Testing, and non-functional testing. |
| It checks whether the software conforms to specifications or not. | It checks whether the software meets the requirements and expectations of a customer or not. |
| It can find the bugs in the early stage of development. | It can only find the bugs that could not be found by the verification process. |
| The goal of verification is application and software architecture and specification. | The goal of validation is an actual product. |
| The quality assurance team does verification. | Validation is executed on software code with the help of the testing team. |
| It comes before validation. | It comes after verification. |
| It consists of checking documents/files and is performed by humans. | It consists of the execution of the program and is performed by the computer. |

**Evolutionary Prototyping**

Evolutionary prototyping also called breadboard prototyping is based on building actual functional prototypes with minimal functionality in the beginning.

By using evolutionary prototyping, the well-understood requirements are included in the prototype and the requirements are added as and when they are understood.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Coding Standards**

* Limited use of globals
* Naming conventions
  + localData (1st latter is capital of every latter other than 1st)
  + GlobalData (camel casing)
  + CONSDATA (all caps)
  + local\_variable ( \_ between every word)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Requirement Traceability Matrix (RTM)**

Requirement tracing, a process of documenting the links between the requirements and the work products developed to implement and verify those requirements. The RTM captures all requirements and their traceability in a single document delivered after the life cycle.