

**CYCLE 2 – SESSION 12**

**TEST-DRIVEN DEVELOPMENT**

**Worksheet 2.1**

**Check out the list of words and tell your teacher and classmates how many of them you already know. Share their meanings with the class and write down the meanings of the ones you don’t know.**

**Vocabulary:**

* Refactoring: recoding, restructuring, rewriting
* Enable: facilitate, permit
* Long Term: extended, prolonged
* Feature: feature, characteristic, property, trait, special feature, peculiarity
* Test Scripts: software testing
* So That: in consequence, for this reason
* Be Bound To: tied to, connected to
* Fail/Failure: be unsuccessful
* Behave: act
* Run: execute
* Enhance: improve, permit

**Worksheet 2.2**

**Match the vocabulary word to its synonym.**

1. **Refactoring**

**(5)** software testing.

**(3)** extended, prolonged.

**(11)** improve, strengthen

**(8)** be unsuccessful.

**(10)** execute.

**(9)** act.

**(1)** recoding, restructuring, rewriting.

**(7)** tied to, connected to.

**(4)** characteristic, attribute, property.

**(6)** in consequence, for this reason.

**(2)** facilitate, permit.

1. **Enable**
2. **Long term**
3. **Feature**
4. **Test scripts**
5. **So that**
6. **Be bound to**
7. **Fail/failure**
8. **Behave**
9. **Run**
10. **Enhance**

**Worksheet 2.3**

**Graphic Organizers**

**a) Match the type of graphic organizer to its definition.**

1. Concept map
2. Main idea web
3. T-Chart
4. Venn diagram
5. Sequence chart

T-Chart:helps organize ideas into two columns and examine two components of an object, concept, or events.

Concept map: shows relationships between the main idea and other information.﻿ Concepts or ideas are represented in circles or boxes and are linked to related ideas with arrows.

Venn diagram: is used to compare and contrast two or more groups of things by visually displaying their similarities and differences in two or more circles that overlap.

Sequence chart: (or flow diagram) presents a series of steps or events in order.

Main ideas web: starts with a central idea and branches out into related ideas and details (or sub-ideas).

**b) Look at the pictures and write the names of each graphic organizer.**

Imagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media

Imagen que contiene objeto, reloj, pelota

Descripción generada automáticamente

**\_\_\_T-CHART\_\_\_**



**\_\_TREE OF IDEAS\_\_\_**

Imagen que contiene tabla, cuarto

Descripción generada automáticamente

**\_\_CHAIN OF SEQUENCES\_\_\_**

Gráfico de burbujas

Descripción generada automáticamente con confianza media

**\_CONCEPT MAP\_**

**\_\_VENN DIAGRAMERS\_\_**

**Worksheet 2.4**

**Read the following text about Test-Driven Development. Then create a graphic organizer that shows the three phases of TDD.**

**What is Test-Driven Development (TDD)?**

Test Driven Development (TDD) is a software development practice that focuses on creating unit test cases before developing the actual code. It is an iterative approach that combines programming, the creation of unit tests, and **refactoring**.

The TDD approach originates from the ***Agile manifesto*** principles and ***Extreme programming***. As the name suggests, the test process drives software development. Moreover, it’s a structuring practice that **enables** developers and testers to obtain optimized code that proves to be resilient in the **long term**.

In TDD, developers start creating small test cases for every **feature** based on their initial understanding. The primary intention of this technique is to modify or write new code only if the tests fail. This prevents duplication of **test scripts**.

**Three phases of Test Driven Development**

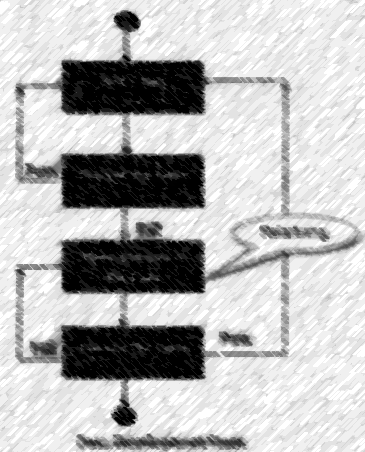
Step 1: Create precise tests. Developers need to create precise unit tests to verify the functionality of specific features. They must ensure that the test compiles **so that** it can execute. In most cases, the test **is bound to** **fail**. This is a meaningful **failure** as developers are creating compact tests based on their assumptions of how the feature will **behave**.

Step 2: Correcting the Code. When a test fails, developers need to make the minimal changes required to correct the code so that it can **run** successfully when re-executed.

Step 3: Refactor the Code. Once the test runs successfully, check for redundancy or any possible code optimizations to **enhance** general performance. Ensure that refactoring does not affect the external behavior of the program.

**Worksheet 2.5**

Share your graphic organizers with the rest of the group. After that, look at the image that your teacher projects on the screen and compare it with your own.

****

**Worksheet 2.12.6**

**Complete the following self-assessment section.**

1. Entiendo qué es Test-Driven Development.

Yes **😃** Maybe **😐** No **😟**

1. Entiendo cuáles son las tres fases de TDD.

Yes **😃** Maybe **😐** No **😟**

1. Entiendo cómo diseñar un organizador gráfico para describir un proceso.

Yes **😃** Maybe **😐** No **😟**

1. Diseñar un organizador gráfico me ayuda a entender un texto mejor.

Yes **😃** Maybe **😐** No **😟**