1. One of the most important things with **Mockito** is to understand
   1. **Why** we’re using **Mockito**?
   2. **What is the context**, **Mockito** fits into?
2. To understand this, we need to set up our **S**ystem **U**nder **T**est (SUT)
   1. We will set up an example that we want to test.
   2. The example we will be using is a **ToDoService**.
   3. **ToDoService** is basically an **interface**.  
      You can think of it as it integrating with some external webservice or MQ or some external system & getting us a list of To-Do
   4. **ToDoBusinessImpl.java** will be using **ToDoService** to get a list of To-Do.
   5. We will be **mocking ToDoService** instead of letting **ToDoService** to talk to some external systemto let **ToDoService**’s methods to return some dummy values.
   6. Using those returned values, we will be testing methods from **ToDoBusinessImpl.java.**
   7. **Briefly:**
      1. Wewant to test methods from **ToDoBusinessImpl.java** but those methods need To-Do data which are being received by those methods using **ToDoService.java’s methods.**But assume that we don’t have any **implementation** for **ToDoService.java** at hand but we want to test **ToDoBusinessImpl.java’s methods.  
         Solution: Mock ToDoService.java’s methods.**
3. Diagram

   Description automatically generated