2.dia:

Until 1997 no one heard of Test Driven Development which asks us to write the tests before we write production code.
TDD is a software development process relying on software requirements being converted to test cases so we write the requirements as special test cases and write the code accordingly so that it will pass the test easily, so the last step is just refactor your code.

3.dia:

Unit test is a type of software testing where you isolate a section of code and verify its correctness.
A unit can be an individual function, method, module, or object.
The goal is to validate that each unit of the software code work as we expected.

4.dia:

Unit testing is the first level of testing before integration testing.
These are typically automated tests written and run by software developers.

5.dia:

Let's talk about the advantages of Unit Testing:

1. Unit tests help to fix bugs early in the development.Its make it easier to find the location of the mistake in the code. Since the unit tests alert the development team of the problem before handing the code to testers or clients, potential problems are caught early in the development process.
2. It can save costs because software developers sometimes try saving time doing minimal unit testing but the not right testing leads to high cost defect fixing after the application is built.
   If unit testing is done in early development, then it saves time and money in the end.
3. As the test is written and run by the developer, it helps to understand the testing code and enables them to make changes quickly.

6.dia:

4. The code will be cleaner and more understandable because unit test provides strict conditions that the piece of code must satisfy. So Unit testing can force developers to structure functions and objects in better ways.
5. Unit tests help with code re-use. You can use your code and also your tests to another project.
6. Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly

7.dia:

Excuses: It requires time, and I am always overscheduled. My code is rock solid! I do not need unit tests.
If you not write unit tests, its lead to a vicious cycle as you can see:
you will not be fast, productive and satisfied enough.
Programmers think that Integration Testing will catch all errors.
The more code you write without testing, the more paths you have to check for errors.
Very simple errors which could have very easily found and fixed with unit test take a very long time to be traced and fixed with another type of tests.
So it's not worth making excuses for the unit test.

8.dia:

Lets talk about the disadvantages:

So i find two disadvantages. The first is is one of the excuses what i mentioned before: It takes time, because with Unit Test, you have to increase the amount of code that needs to be written.
Unit testing can't be expected to catch every error in a program. There is no way it can test every execution path or find integration errors and full system issues.

9.dia:

Clean tests follow five rules:
F Fast When tests run slow, you won't want to run them frequently and then you won't find problems early enough to fix them easily.

I Independent Tests should not depend on each other.
You should be able to run each test independently and run the tests in any order you like**.**
R Repeatable Tests should be repeatable in any environment.
If they aren't then you'll always have an excuse for why they fail.
S Self-Validating The tests should have a boolean output either they pass or fail.
You shouldn't have to read through a log file or manually compare two different files to see whether the tests pass.
T Timely Unit tests should be written just before the production code.
If you write production code first, then you may find that to be hard to test or you may not design the production code to be testable - because you don't know what you want to test exactly.

10.dia:

Tests are as important for the success of a project as the production code is.