Self Introduction

Nikunj Majmudar - 98795-80089
nikummajmudar@yahoo.in

What is Software Testing?

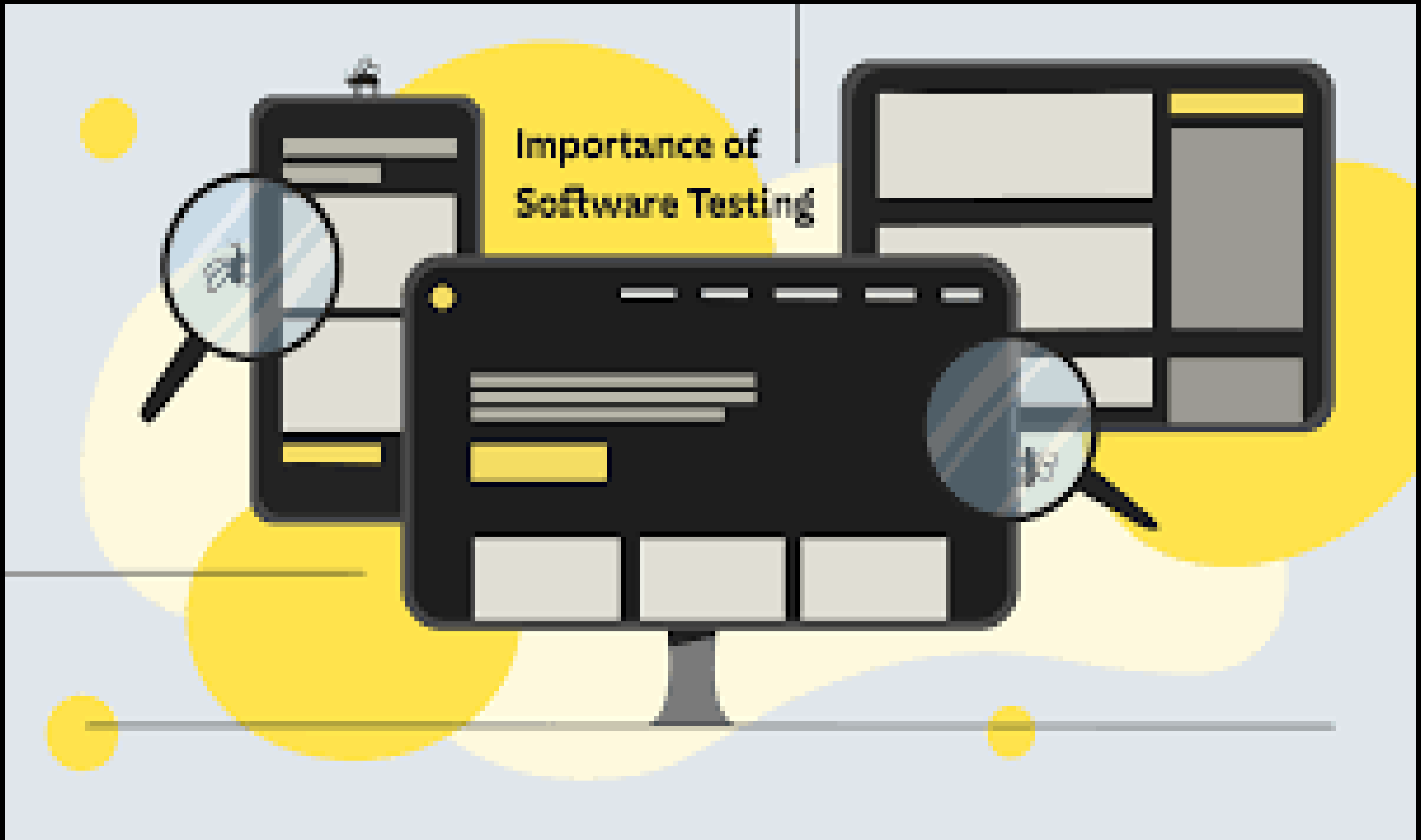
- Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not.
- In simple words testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements .
- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

Why testing is necessary?

- Testing is necessary because we all make mistakes.
- Some of those mistakes are unimportant, but some of them are expensive or dangerous.
- Ideally, we should get someone else to check our work because another person is more likely to spot the flaws.
- Software Systems are now part of our everyday life

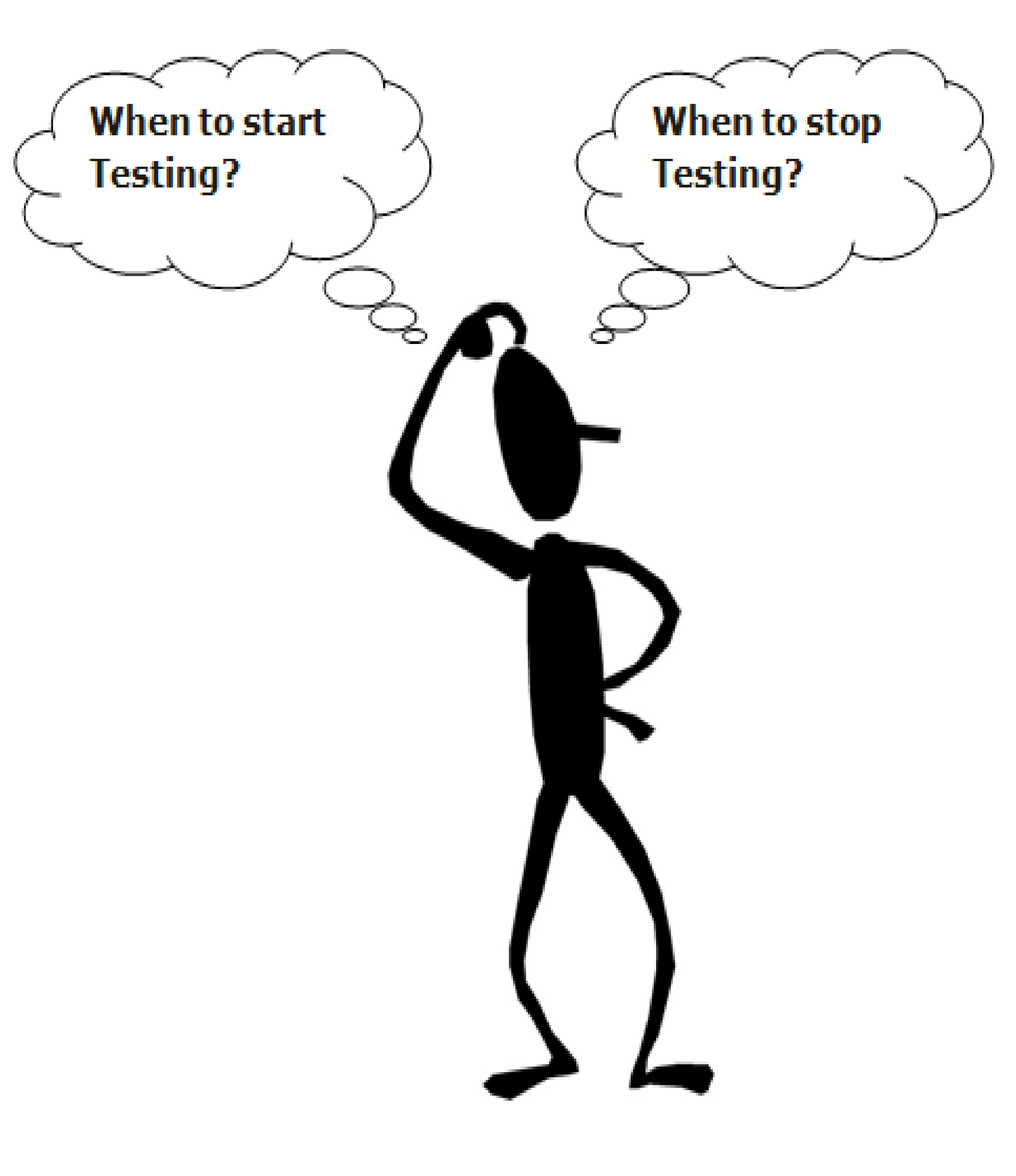
- For example :
- Banking and Financial institutions, Retail industry Central and Local Government, Transport (e.g. Planes, Trains and Automobiles), Medicine (Hospitals, research centers), Home Entertainment,
- We have all experienced Software Systems failing!

- Software System Failures can lead to: Human Injury or Death , Technological disasters, Loss of face for suppliers and/or their customers; Legal action and associated costs.



When to start testing?

- Testing is sometimes incorrectly thought as an after-the-fact activity; performed after programming is done for a product. Instead, testing should be performed at every development stage of the product.
- If we divide the lifecycle of software development into "Requirements Analysis", "Design", "Programming/Construction" and "Operation and Maintenance", then testing should accompany each of the above phases.
- If testing is isolated as a single phase late in the cycle, errors in the problem statement or design may incur exorbitant costs.
- Not only must the original error be corrected, but the entire structure built upon it must also be changed.
- Therefore, testing should not be isolated as an inspection activity. Rather testing should be involved throughout the SDLC in order to bring out a quality product.



When to stop testing?

- "When to stop testing" is one of the most difficult questions to a test
- engineer. The following are few of the common Test Stop criteria:
- All the high priority bugs are fixed.
- The rate at which bugs are found is too small.
- The testing budget is exhausted.
- The project duration is completed.
- The risk in the project is under acceptable limit.

Seven key principles?

1. Testing shows presence of Defects

2. Exhaustive Testing is Impossible!

3. Early Testing

4. Defect Clustering

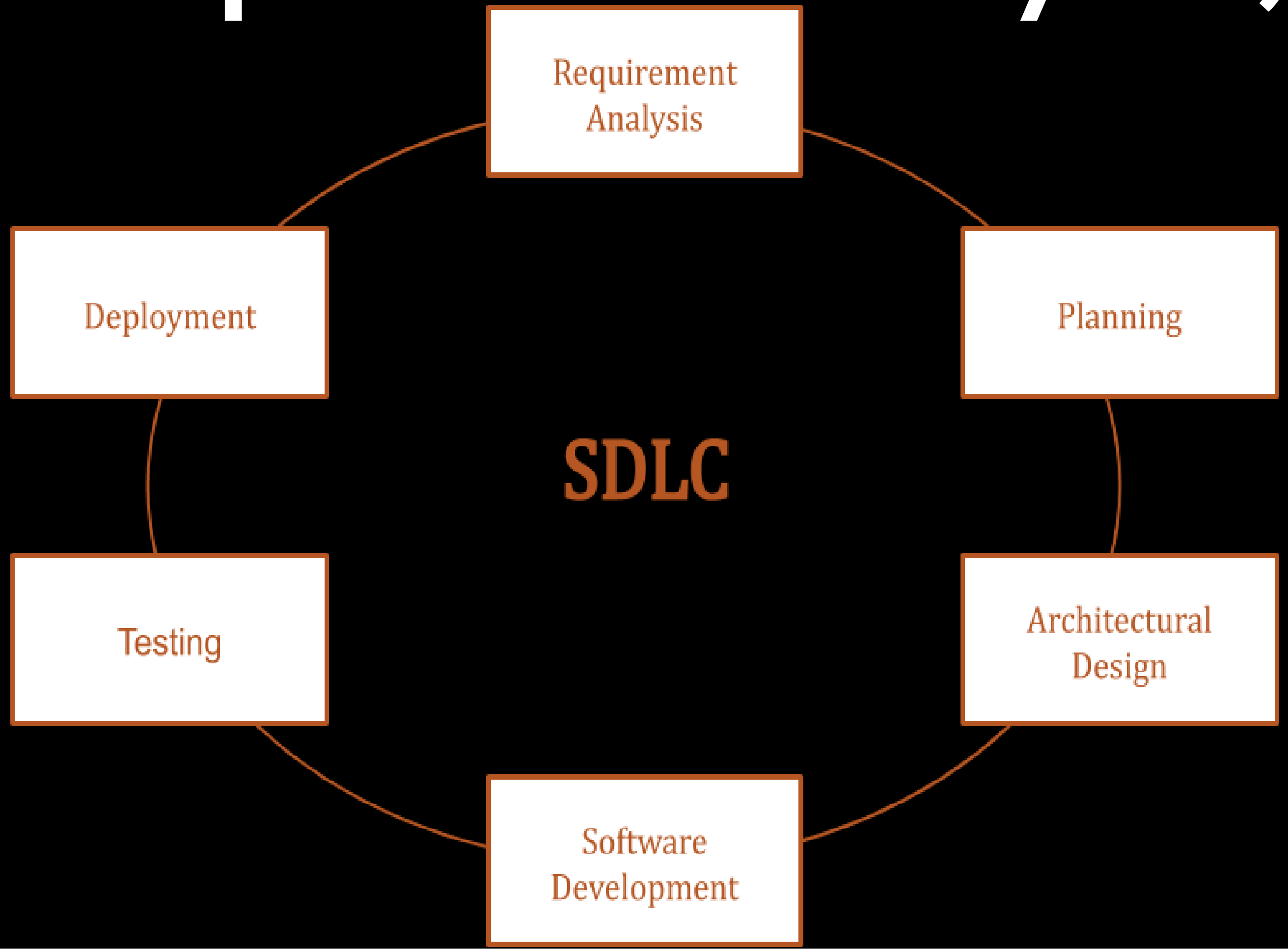
5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy

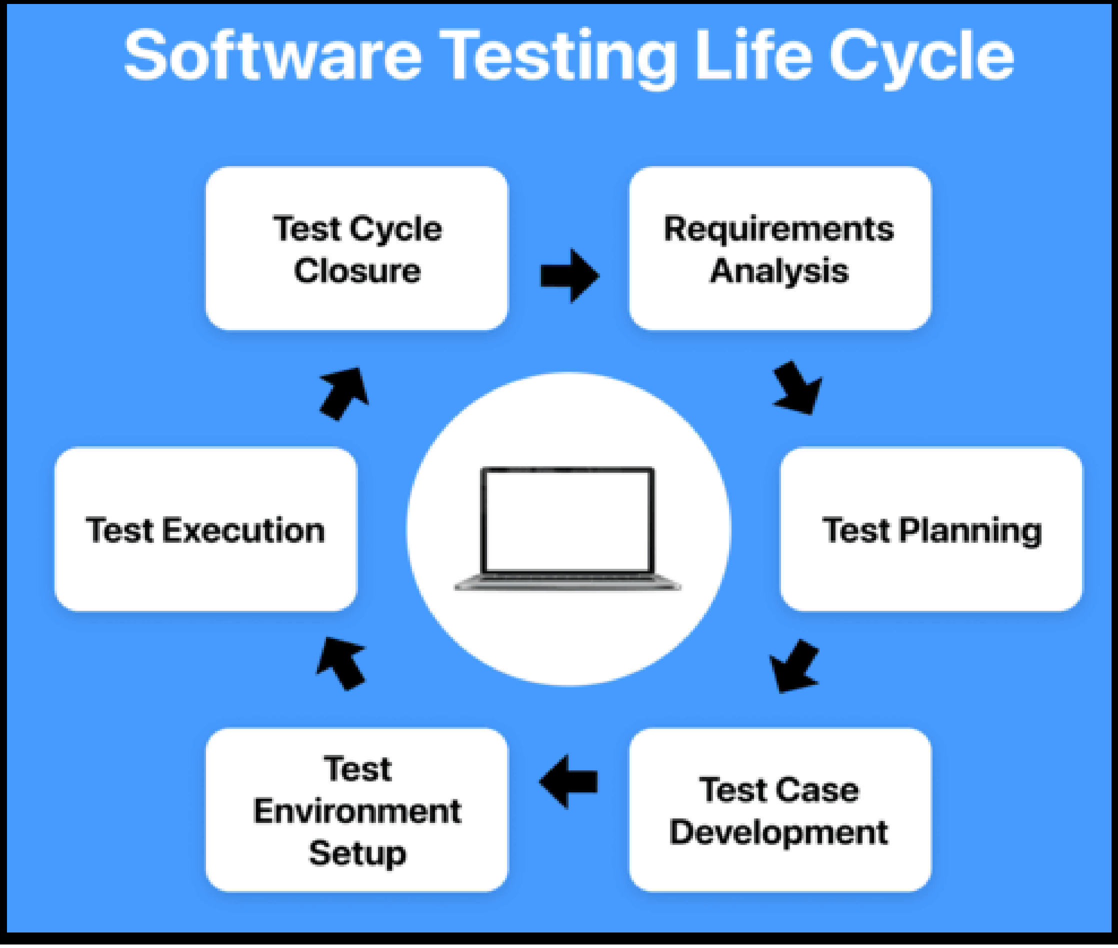
SDLC (Software Development Life Cycle)?

1. **Requirements Collection / Gathering**
2. **Analysis**
3. **Design**
4. **Implementation**
5. **Testing**
6. **Maintenance**



STLC (Software Testing Life Cycle)?

1. Requirement Analysis
2. Test Planning
3. Test case development
4. Test Environment setup
5. Test Execution
6. Test Cycle closure



Thank You