Software Development Cycle Workbook

Software Development Lifecycle (SDLC)

The Software Development Life Cycle (SDLC) is a process used by the software industry to produce high-quality systems that meet or exceed the customer's expectations.

- Works effectively (performs its task well).
- Works efficiently (does not cost too much to operate).
- Is inexpensive to maintain.
- Is cost-effective to enhance.
- Must be produced within time and cost estimates.

Phases of SDLC

1. Business/Requirement Analysis
2. Design
3. Development/Coding
4. Testing
5. Deployment
6. Maintenance

1A— Business Analysis	● One bedroom apartment (we now have 2 kids). ● Limited closet space. ● High rent (more than a mortgage). ● Small kitchen (would like more space). ● One bathroom servicing 4 people. ● No parking space. ● No privacy. ● No place for the kids to play. ● Too many people in neighborhood.
1B— Requirements Analysis	
2— Design	• High Level Design (HLD) - lists the functional aspects of the various modules along with the final result. • Low-Level Design (LLD) - details the logic and execution of each module in an HLD.
3— Development Coding	• Now that we have a clear understanding of how the application is supposed to work, it is time for our team of programming experts to begin coding. • Programming tools like compilers, interpreters, and language such as COBOL, C, C++, and Java are used for coding with respect to the type of application.

4— Testing	• **Software Quality** is best determined by how well the software meets the requirements.
	• Types of tests
	  ○ User Experience Testing
	  ○ Functionality Testing
	  ○ Security Testing
	  ○ Load Testing
	  ○ Performance Testing
	  ○ Compatibility Testing

5— Deployment	• Deployment moves the product/service into the "live" environment. Once deployed successfully, customers and users can be allowed to begin utilization of the functionality provided.
6—Maintenance	The Maintenance phase is an ongoing process. • Maintenance of Hardware/Software. • Updates. • Upgrades. • New Features.

High-Level vs. Low-Level

High-level Design	High Level Design (HLD) - lists the functional aspects of the various modules along with the final result.
Low-level Design	High Level Design (HLD) - lists the functional aspects of the various modules along with the final result.

Types of Testing

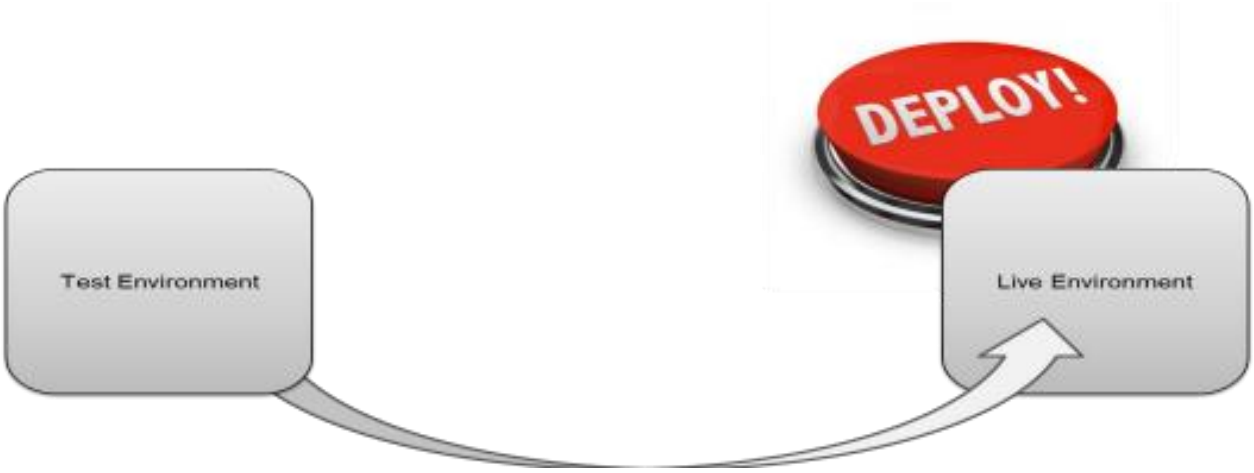
User Experience Testing	How easily can users utilize this functionality?
Functionality Testing	Does this app perform the work needed as designed?
Security Testing	Does this release meet the customer's acceptable risk profile?
Load Testing	Will this service be able to support (x) number of users and maintain acceptable levels of performance?
Performance Testing	Will performance in a live environment meet the planned performance from the blueprint?
Compatibility Testing	What effect on the live environment will this service have once it is released? Will any other live services be adversely impacted by this new introduction?

What is the importance of ensuring users are properly trained?

So users use the application appropriately as built so they don’t break it.

Why does this training need to occur before rollout?
So the people maintaining and using the application know how to utilize and maintain it.

What is meant by “deploy into the live environment''?
Putting the application from the test environment to the live environment so the Operations team can release to the public.



What steps are necessary after deployment into the live environment to allow users to utilize the new functionality?
The Operations team has to “release” it to the customers/users after the training has been completed for them to utilize.