**Henri Siyapdje**

**Software Development Cycle Workbook**

Software Development Lifecycle (SDLC)

| Refers to the methodology for creating software quality.  Most common SDLC are:  > Waterfall model:every phase has to be completed before the next phase starts.  > Agile model: |
| --- |

Phases of SDLC

| 1.  Analysis |
| --- |
| 2.  Planning |
| 3.  Design |
| 4.  development/Coding |
| 5.  Testing |
| 6.  Deployment |

| **1A— Business**  **Analysis** | Identify the current State, where are we now?  Understanding the current state allows us to identify the needs that need to be fulfilled by our project. |
| --- | --- |
| **1B— Requirements Analysis** | **Entry Criteria**: Requirements document available (functional and non-functional).  **Activity**:Know the business modules and modules-specific functionality.  Identify all transactions in the modules and also the user profiles  Gather user interface/authentication and geographic-spread requirements.  Identify types of tests to be performed and gather testing priorities and focus.  Prepare RTM(Requirement Traceability Matrix),identify the test environment.  **Exit Criteria:signed off RTM and Test automation feasibility** |
| **2— Design** | HLD(High Level Design)  LLD(Low Level Design) |
| **3— Development/ Coding** | What language to use ?  Example: Java, Python, Ruby, groovy,c++ |
| **4— Testing** | STLC(Software Testing Life Cycle)  > Entry criteria (requirement Traceability Matrix, Test automation feasibility report)  > Exit Criteria ( approved Test plan, signed-off effort estimation document)  Requirement Analysis(Analysis, planning, test case, test environment, execution, Closure)  Regression Testing : (Software testing) retesting to make sure that there are no bugs. |
| **5— Deployment** | Automated tools are used here to ensure that the software are is deployed securely,  Firewalls, access controls, and security settings are also configured. |
| **6—Maintenance** | Hardware and Software should be maintained after the Closure phase. So, they periodically need Update for bugs fix and upgrade for new feature |

| High-level Design | List the functional aspects of the various modules along with the final result. |
| --- | --- |
| Low-level  Design | Detail the logic and execution of each module in HLD |

| User  Experience  Testing | Users must be trained in how to interact with the new systems prior to the new rollout.  Not having users ready to interact with the system is a major risk to a successful implementation. |
| --- | --- |
| Functionality Testing | Requirement type considerations  Functional: define what the software must accomplish.  Non-functional: define availability, capacity,continuity, and security. |
| Security  Testing | Does this release meet the customer ‘s acceptable risk profile? |
| Load  Testing | The service should be able to support the number of users and maintain acceptable levels of performance. |
| Performance Testing | Will the performance in the production meet the planned performance from the blueprint. |
| Compatibility Testing | Will any other liveservices be adversely impacted by this new introduction. So a compatibility check should be performed. |

What is the importance of ensuring users are properly trained?

trained users understand the Software development life cycle.

They adhere to best practices, reducing errors, bugs and security vulnerability

Also they can identify potential risk early

They are more likely to provide valuable feedback.

Why does this training need to occur before rollout?

Training users before the rollout sets the stage for successful implementation,user adoption, and project success.

What is meant by “deploy into the live environment''?

This marks the transition from development and testing phases to actual use by customers or stakeholders.



What steps are necessary after deployment into the live environment to allow users to utilize the new functionality?

First of all, Communication and training to make sure that the user understands how to use it. Then,

User Acceptance Testing encourages users to perform UAT on the new functionality to verify that it meets their requirements and expectations in the live environment. Also update user documentation to reflect the new functionality. In addition, put in place a mechanism to get users feedback. Continuously monitor the performance and usage to identify any potential issues