**IV. Diving into UAT BAU & Deliverables**

**Topics**

* Test planning & strategy – Outlining Test plan
* Analyzing requirements – Designing Requirement Traceability Matrix
* Test design – Designing Test cases, Review / Query log
* Test execution & Defect reporting – Fabricating Daily Status Reports, Pass Logs & Defect Logs
* Sign off

**Test planning – Outlining Test plan**

A test plan is a detailed document that outlines the test strategy, Testing objectives, resources (manpower, software, hardware) required for testing, test schedule, Test Estimation and test deliverables.

The test plan serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

Before UAT cycle starts, a Test plan is created to guide the entire UAT cycle.

Making Test Plan has multiple benefits

Test Plan helps us determine the effort needed to validate the quality of the system.

Help people outside the test team such as developers, business managers, customers understand the details of testing.

Test Plan guides our thinking. It is like a rule book, which needs to be followed.

Important aspects like test estimation, test scope, Test Strategy are documented in Test Plan, so it can be reviewed by Management Team and re-used for other projects.

**Test Strategy**

Depending on the timelines and scope of Testing, a strategy is devised to cope up with the deadlines.

A Test Strategy is a plan for defining the testing approach and it answers to questions like what you want to get done and how you are going to accomplish it.

One of the key challenge in UAT is time crunch and coping up with deadlines so that the system goes into production on time but at the same time preventing any defect leakage. Thus it is important to prepare a strategy to achieve this.

Priority / Risk based is one of the most prevalent used Test Strategy

Risk based testing is basically a testing done for the project based on risks. Risk based testing uses risk to prioritize and emphasize the appropriate tests during test execution. In simple terms – Risk is the probability of occurrence of an undesirable outcome. This outcome is also associated with an impact. Since there might not be sufficient time to test all functionality, Risk based testing involves testing the functionality which has the highest impact and probability of failure.

A list is prepared with all functionalities. A risk number is calculated with each of them. Risk number is calculated by taking product of Likelihood and impact of failure. Higher the number, high risk associated with the functionality. In simple terms, if that particular functionality fails, it will cause a huge impact.

Thus functionalities are tested in order of their risk numbers starting from areas with highest risk numbers.

**Analyzing requirements**

Requirements are shared either in a word document (known as Business Requirements Document) OR as user stories in JIRA tool.

Reading and understanding the thoroughly is required. A tester must not assume anything while understanding them. When in doubt raise a flag.

The skill here is identify high level scenarios for each requirement in way that accomplishes 100% coverage of entire system wrt testing.

**Requirements Traceability Matrix** is prepared to ensure this. A list of requirements is prepared and the corresponding test scenarios for them are noted thus mapping them. This will easily highlight any requirement which is not covered by any Tests.

**Test Design**

On the basis of requirements, Test cases are designed. Test cases must be written in a way that any third person new to the system must be able to execute them.

Test cases include detailed steps with expected and actual results for each step.

Test cases can be designed in an excel spreadsheet as agreed upon by stakeholders.

Test cases need to be reviewed and signed off by business before using them for Test execution.

**Understanding a Test Case document**

**Review logs**

As mentioned, Test cases need to be reviewed by business. Incorrect test cases have the potential to cause huge losses in production.

For this process, a predefined format of review logs is used. This format includes list of all Test cases and to whom they are assigned for review, status, comments. It needs to be updated every day in Test Design phase. All points in the review logs need to be in closed status i.e. reviewed by respective stakeholders and signed off by them.

**Test Execution & Defect Reporting**

Once the system is ready to be tested, tester starts execution the test cases designed. Test cases designed are executed by comparing actual results with expected results.

Actual results are recorded in the Test case spreadsheet along with screenshots which are used as evidences. These screenshots are nothing but Pass Logs.

**Defect Logs**

Any discrepancy between expected and actual result is reported as a bug. These are then reviewed upon pushed for fixing.

Bugs / defects are reported in JIRA or in the below format known as Defect log:

A defect log must clearly define the defect and detailed steps to be executed to reproduce the bug. A screenshot to each defect is mandatory.

**Daily Status Reports**

In the entire Test execution cycle of UAT, a concise report needs to be sent out to all stakeholders at the end of each day’s execution.

This report captures test execution status, progress, challenges, RAG status, defect status and different metrics.

**Sign off**

* Upon successful completion of Test execution, a sign-off email is sent out which states whether the system is good to be deployed into production.
* It consists of summary of Test cases executed, Test cases passed, failed, Defects and their respective status, severity, etc.
* On the day of sign off, all stakeholders catch up to decide whether the system is stable enough and ready to be deployed in production or not. This is commonly known as a go/ no-go call.
* The decision is taken by business leveraging the test results, open defects, etc. parameters.

**Conclusion**

In this session, we got insights on UAT phases and the corresponding key deliverables in each phase.

These deliverables & documentation are specific to the project and the SDLC model used.

The documents and its format changes accordingly as agreed upon by the stakeholders.