

Experiment : 04

AIM: Preparation of Software Testing phase document of some problem.

Theory:

Testing Phase Testing

1. Introduction

This approach document is designed for the Information and Technology Services System Development Life Cycle (SDLC) projects.

A. Purpose

The intent of Development/Unit Testing is to efficiently and accurately develop, unit test, and provide documentation for Designs produced. Repeatable, consistent processes and comprehensive documentation support this goal. The end result should be increased quality (fewer bugs), and the reduction of delivery time and effort needed for subsequent projects.

2. Approach

A. Approach Description

The general methodology/approach for SDLC Development/Unit Testing at ITS will involve the following steps:

1. Initial planning efforts for this phase will include the establishment of:
 - a. A project status reporting approach.
 - b. A defect tracking approach.
 - c. A metric collection approach for the phase.
2. (If New Hardware or Configuration) Perform SDLC Environment Setup & Validation Process (9000)
3. The guidelines for Development and Unit Test work will be:

- a. Eliminate Objects to be obsoleted. System Testing will be the only verification of success for these changes.
 - b. For Objects with Design Documents (S300 — Design Document) the following steps are performed:
 - i. Code the Design.
 - ii. Develop the Unit Test Plan.
 - iii. Review Code and Unit Test Plan if functionality has been added or changed.
 - iv. Unit Test the Coded Objects.
 - v. Review and Approve the Unit Testing results.
4. The order in which the Objects should be Developed and Unit-Tested will be based upon all of the following:
 - a. Priority assigned.
 - b. Grouping of all objects for a Business Process.
 - c. Direction of the SDLC Lead or Project Manager.
 - d. Ordering of System Testing and Acceptance activities.
 - e. Ordering of Objects from Design Phase.
 - f. Availability of Test Data.
5. Steps involved in Coding are as follows:
 - a. Review and understand the information in the S300 Design Document.
 - b. Determine exactly which objects will need to be changed and update the S300 Design Document to include that information.
 - c. Notify any other ITS units affected by the changes made to the S300 Design Document.

- d. Code the Object using the PS Programming Standards and Guidelines in Lotus Notes. Also refer to the Development and Unit Test Checklists as informational guidelines.
- e. Notify the Project Manager of any needed changes or additions to the PS Programming Standards and Guidelines. An interim working standard will be developed to meet this need.

6. Steps to develop an S404 Unit Test Plan for each S300 Design Document:

- Define the conditions to be tested based on functional requirements and include conditions that exercise every path of new or changed logic.
- Follow the Unit Test Process Guidelines in Lotus Notes.
- Refer to the Upgrade Unit Test Checklist as an informational guide in the creation, execution and review of the Unit Test Plan.
- Organize testing into cycles to simplify troubleshooting and analysis of results.
- Prepare the test model with input data and expected results.
- Determine and document test data.
 - o Test good data and boundary conditions.
 - o Ask for Designer's assistance if needed when identifying and/or creating data needed for testing.
 - o Follow Test Data Creation Guidelines in Lotus Notes.

7. Conduct Code and Test Plan Review.

- Attendees could include: Tech Lead, Peer, and Designer.
- Verify the following of Programming and Unit Testing Standards and Guidelines found in Lotus Notes.

- Utilize the Software Development and Unit Test Checklists as informational guidelines during reviews.
 - Update the Unit Test Plan with the appropriate Signoff/Approval signatures.
 - If the Code and/or Test Plan is not approved, make necessary changes and conduct a re-review.
8. Execute the Test Plan.
- Check the output against the expected results
 - Evaluate any unexpected results
 - Make sure that any required corrections are carried out and re-tested
 - Make sure that final testing components (conditions, input, and expected results) are accurate, complete and documented in such a way to make them repeatable and re-usable.
 - Document results by updating the S404 Unit Test Plan
9. Review and obtain signoff of Unit Test results, updating the S404 Unit Test Plan with the appropriate Signoff/Approval signatures.
10. Update the Tracking Spreadsheet/Database.
11. Migrate the Object(s) to the System Test environment in a planned release.

3. Status and Reporting

A. Defect Tracking Approach

During Development/Unit Testing, problems encountered will be tracked within the Tracking Spreadsheet/Database.

B. Time Tracking & Reporting

The Project Manager will be responsible for ensuring that time is entered into PlanView and that Monthly Status Reports are created.

C. Metrics To Utilize

Metrics that will be tracked include but are limited to:

1. Percentage of Objects with Unit Testing complete
 - a. From Planview or the Tracking Spreadsheet/Database.
 - b. Also include interim milestones by objects.
2. Effort Completed as Percentage of Estimated Effort.
 - a. From Planview — Time Reported and ETC (Estimated Time to Completion) updates.
3. Percentage of Effort Earned. (Actual vs. Planned work completed).
4. Percentage of Effort Remaining — only 100% completion factored in.
5. Effort Rate
6. Burn Rate
7. Green/Yellow/Red tasks.

D. Status/Issues Meetings

During Development/Unit Testing, it is recommended that a weekly status meeting be held at the same scheduled time on the same day each week. This meeting will support a communication process to inform the project team of status, problems or issues, and changes. The Agenda will include a review of open high and medium issues, Unit Test plans, and any action items from the previous week. Meeting minutes should be taken that include a list of attendees, discussion points, and action items. TIO/AIG, Module Leads, Data Delivery, Access Services groups etc. should be represented in the meeting, as needed.

E. Acknowledgement

Where deliverables require “acknowledgement”, this means that the responsible party has reviewed the documentation and feels that the documentation was sufficient. It is important to define who is responsible for reviewing and approving any results or documentation early in the project lifecycle.

4. Assumptions and Risks

A. Assumptions

The following assumptions are critical to the successful accomplishment of this Approach to Development/Unit Testing:

- n An S404 Unit Test Plan is created for each S300 Design Document.
- n If the Design included cross-functional input and review, then the unit test should also include cross-functional input and review.
- n Both Technical and Functional knowledge resources should have input to test plan and results review.
- n The Designs should be referenced with the Unit Test Plan through document naming and repository standards as a minimum.
- n Test Plan and Results Signoff may be different for each team. A provision should be made for Reviewer and cross-functional Review if needed.
- n Database tuning help will be available if needed.
- n Retesting will occur after tuning.

This approach makes the following assumptions for the System Test Phase: n Testing will be conducted for each Business Process.

- n Reusable test conditions will be utilized, which may include the use of reusable test scripts.

- n Every business condition will be tested.
 - o Testing will be especially detailed in areas having new functionality.
- n Regression and integration testing is covered.
- n Data plans will be included in the System Test Plans.

B. Risks

- n Completion of Design Phase misses targeted schedule.
- n Insufficient resource levels.
 - Production Support activities pull resources away.
 - Design Phase activities pull developers away.
- n Infrastructure Changes that impact work processes.
- n Tasks underestimated for effort and/or duration.

5. Deliverables

A. Deliverables from the Project Manager

The following deliverables are required.

- Updated Development/Unit Test Phase Approach Document, if needed.
- Updated Metrics/Tracking Reports
- Work Breakdown Structure (WBS)/Schedule - updated tasks and adjusted hours. This is required at the beginning of the Development/Unit Test

Phase, in order to put the WBS into PlanView for tracking and reporting.

B. Deliverables from the Phase

The following deliverables are required from the Development/Unit Test Phase for it to be considered completed.

- S404 - Unit Test Plan
- Unit Test Results
- S405 - Dev/Unit Test Phase Completion Checklist
- A coded, Unit-Tested Application

C. Deliverables from the Methodology Team

The following deliverables are required.

- Updated Process Flows, Templates, or Tools if necessary, based upon project team feedback regarding successes and challenges.

Appendix

A. Definitions

| Test Phases | Focus |
|-------------------------|---|
| Unit Test | Testing individual modules and programs, and testing them in sufficient context to <u>insure</u> that work flows correctly through the affected business process. |
| Integration/System Test | Validates that all processes, including customizations and interfaces, work together to support the business functions |
| Volume/Stress Test | Validates that critical functions will meet production performance requirements |
| User Test | Validating production-ready functionality and data integrity |
| Regression Test | Ensures that the application doesn't negatively impact previously migrated objects/modules. Re-tests the application to ensure that the application performs correctly after a package upgrade or change. |

Result: The Testing Phase Document is prepared.