

# Unified Functional Testing

Introduction to Test Automation

## Lesson Objectives

By the end of this Lesson you will be able to:

- Understand the purpose and use of test automation
- Identify good candidates for test automation
- Understand the life cycle of test automation



## Lesson Content

1. What is Test Automation?
2. Test Automation Highlights
3. What to Automate
4. Automation VS Manual
5. Test Automation Life Cycle
6. Project Lifecycle Align With Test Automation
7. Test Automation Packages



# Test Automation-Definition

**“The use of software to control the execution of tests, the comparison of actual outcomes to predicted outcomes, the setting up of test preconditions, and other test control and test reporting functions”.**

*(TekSci's Dictionary)*

- Test automation is mainly used for functional regression testing

## Test Automation Highlights

- ▶ Speed up the testing process
- ▶ Repetitive scenario insured
- ▶ Achieve more reliable tests by minimizing human error
- ▶ Higher quantity and quality of test cases
- ▶ Using Automation tool at night for utilizing your testing effort very efficiently



## Test Automation Initiation Conditions

- ▶ ROI analysis proves worthiness
- ▶ Stability of the AUT (Application Under Test)
- ▶ Knowledge of the manual procedures to be automated



## Which Test Cases To Automate?

- ✓ Tests that need to run every build of the application (Sanity Check, Regression Test)
- ✓ Tests that use multiple data values for the same actions (data driven tests)
- ✓ Tests that require detailed information from application internals (e.g., SQL, GUI attributes)

*The more repetitive its execution – the better a candidate for automation!*



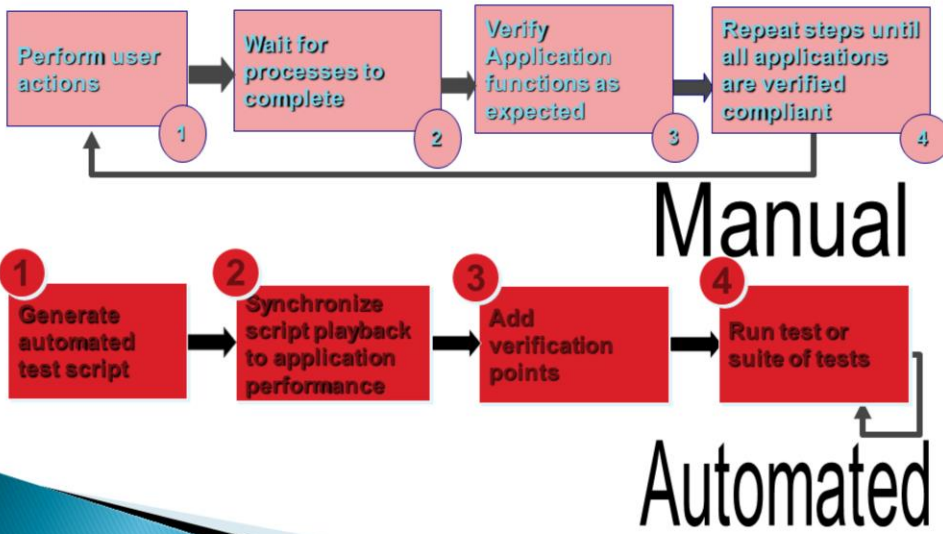
## Which Test Cases Not to Automate?

- ✗ Usability testing
  - "How easy is the application to use?"
- ✗ One-time testing
- ✗ "ASAP" testing
  - "We need to test NOW!"
- ✗ Ad hoc/random testing
  - Based on intuition and knowledge of the application
- ✗ Tests without predictable results

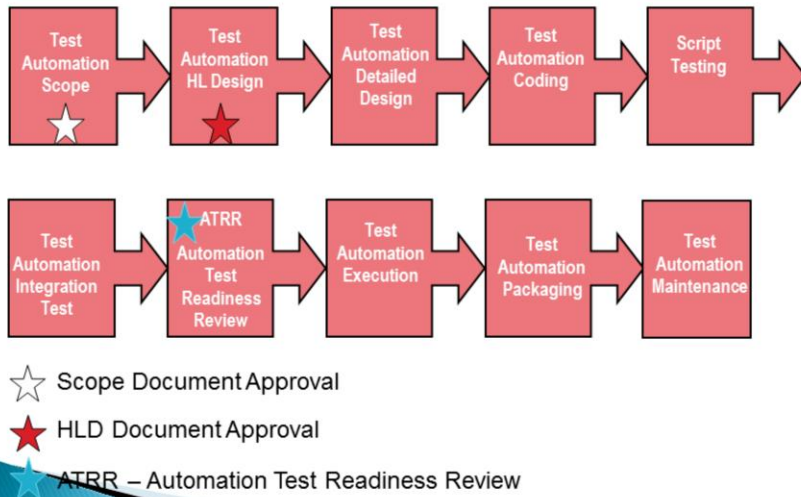
*Improvisation required? Poor candidate for automation*



## Manually Vs. Automated

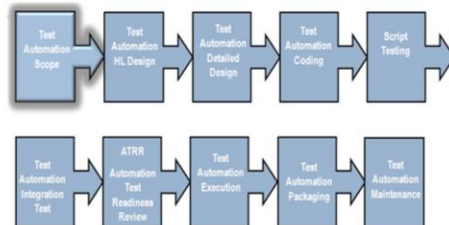


# Test Automation Life Cycle



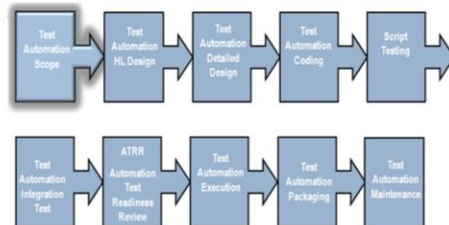
## Test Automation Life Cycle – Scope Considerations

- ▶ SW modules that meet the automation worth criteria
- ▶ Available professional resources
- ▶ SW maturity
- ▶ Integration between the automated tool and the System Under Test (SUT)



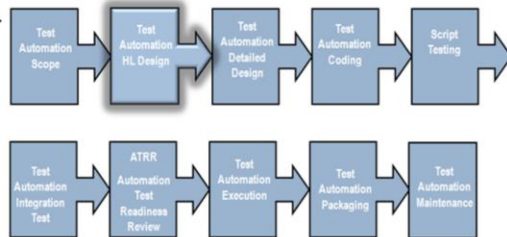
## Test Automation Life Cycle – Scope Guidelines

- ▶ Needs to be run every build
- ▶ Multiple data for the same business activity
- ▶ Test with predictable results
- ▶ Regression tests are most commonly covered
- ▶ Priorities in the scope
- ▶ Based on ROI Model



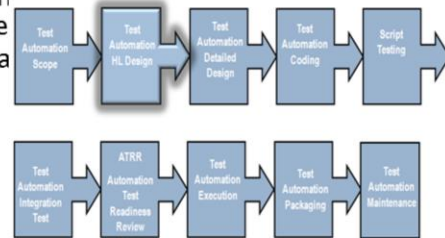
## Test Automation Life Cycle – High Level Design

- ▶ The HLD includes the following:
  - Design the activities within the scope (in TD AL) – with a detailed expected results
  - Define Scenarios (Calendars)
  - If applicable, re-use and maintain existing automated solutions
  - Resources & Timeta



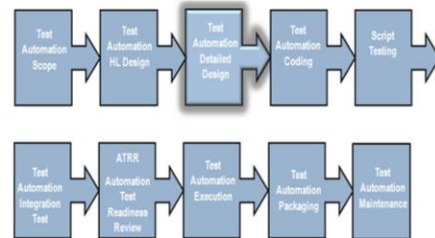
## Test Automation Life Cycle – High Level Design – Cont.

- ▶ Description of the Testing Tools environment includes detailed explanation regarding the tool usage and versions
- ▶ Requirements for the Test Automation Development & Execution environment
- ▶ Deliverables –
  - HLD Document.
  - QC Manual Activity Library with Test Cases procedure
  - Designed Calendars in Calendar application.



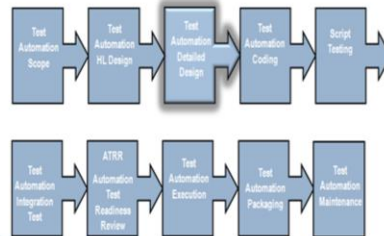
# Test Automation Life Cycle – Detailed Design

- ▶ The DD includes the following:
  - Global scripts and environment configuration
  - Infrastructure function (eg ATL) preparations
  - Test Data structure preparations
  - Test automation environment preparations



## Test Automation Life Cycle – Detailed Design – Cont.

- ▶ For every test case, within the scope, define the following:
  - Prerequisite
  - Script Parameters
  - Error handling and recovery mechanism
  - Internal script algorithm (Structure and Flow)
  - Scripts dependences
  - Data table structure and content
  - Verifications list
- ▶ Deliverables –
  - DD Document.

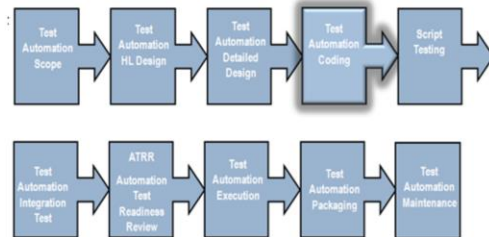




## Test Automation Life Cycle – Coding

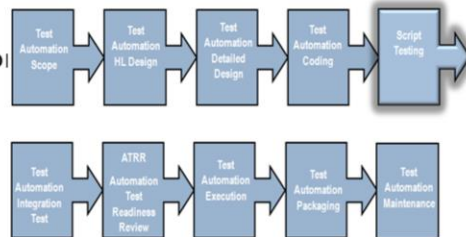
- ▶ The automation development phase is done with UFT according to the Automation Standards & Guidelines and includes the following:

- Skeleton Scripts – recorded and/or coded
- UFT Coding Standards (Documentation, Naming Conventions, Declarations, Functions)
- Error Handling
- Synchronization Points
- Verification Points
- Common Data Files
- Compiled modules
- Reporting Results



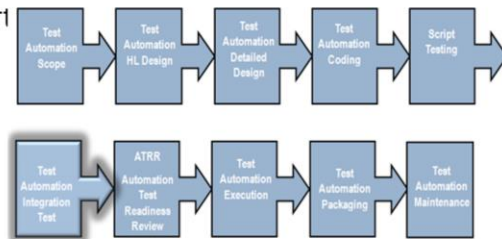
# Test Automation Life Cycle – Scripting Testing

- ▶ Script testing should be performed for each test script/function separately
- ▶ Each script is unit tested in UFT debug mode
- ▶ Scripts are tested with all relevant input data options
- ▶ The execution report is reviewed
- ▶ Both negative & positive cases should be checked
- ▶ Deliverables –
  - Verified scripts/functions
  - Well-defined execution repository



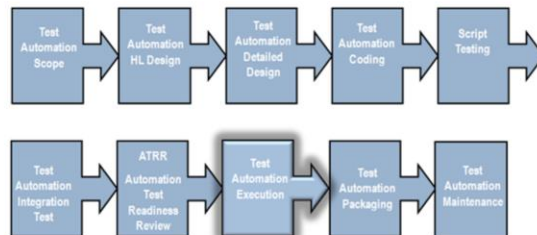
## Test Automation Life Cycle – Integration Testing

- ▶ Check the integration between the scripts (eg scripts that share the same data) to make sure that the test set runs as one suite
- ▶ Support intermediate failures (this is important for the overnight run)
- ▶ Verify test environment
- ▶ Integration testing execution is done from Test Director's Test Set (after Calendar export)
- ▶ Review entire results report
- ▶ Deliverables –
  - Verified modules
  - Verified function libraries



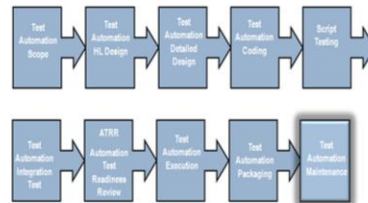
## Test Automation Life Cycle – Execution

- ▶ Run the test set in a dedicated environment
- ▶ Run the Test set from QC after export with the real data assignments
- ▶ Deliverables –
  - Defects report
  - Execution report



## Test Automation Life Cycle – Maintenance

- ▶ Two main maintenance activities:
  - Enhance the Test Automation depth
  - Adjust the Test Automation package to the changed SW
- ▶ Every maintenance activity should be treated as a development process (depending on its size) and should follow all the life cycle phases.



# Freeze Procedure

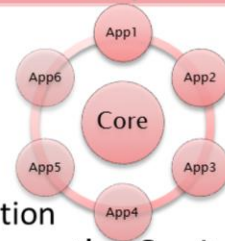
- ▶ When closing an automation project version perform the following activities:
    - ATRR
    - Prepare Execution User Guide in Document Center after technical writer review
    - Input Excel files – Only the active sheet should remain in each Excel
- All other 'debugging' sheets should be deleted
- Quality Center Test Cases:
    - Attach the Excel files of each script to the test script
    - Attach the relevant GUI files to the Init script
    - Attach all the documentation to the Test Set folder, in the design phase
  - Backup the version via the Help Desk

# Test Automation Packages – Automated Sanity Test

***Set of ~50 UFT scripts that verifies that the new build is ready for massive testing***

*Sanity Test*

- GUI functional testing
- Main processes validation
- The basic functionality in each application
- ▶ Once a SW build successfully passes the Sanity Test, it is ready for more extensive functional and non-functional testing

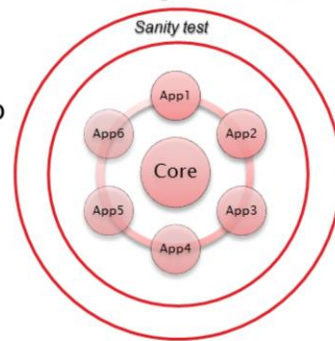


# Test Automation Packages – Automated Mini Regression

***Set of ~110 UFT scripts that verifies that all the existing functionality have not been affected by the current version***

- Based on the sanity check package (Scope and Architecture)
- Include additional scope according to the test automation worth criteria (breadth and depth)

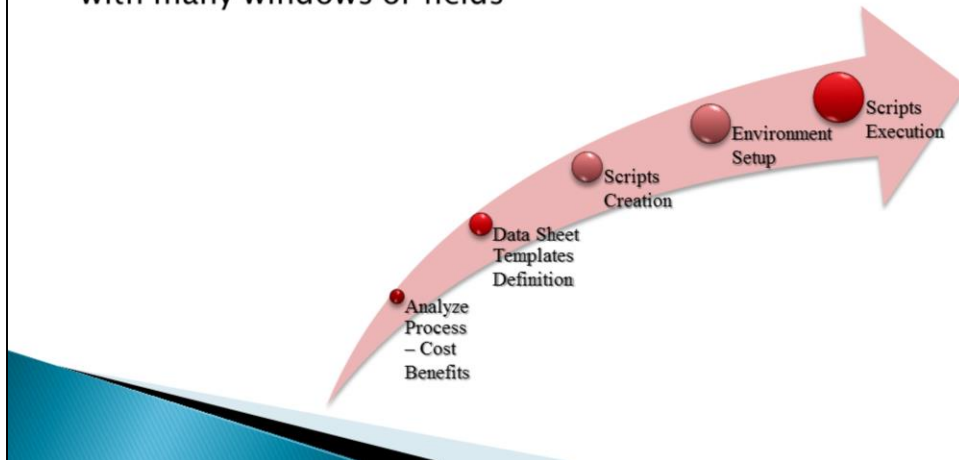
*Mini Regression Test*





## Test Automation Packages – Automated Data Inflation

- This package uses UFT to record the most frequent on-line activities and/or those that involve manual work with many windows or fields



## What's Next?

- Review Questions
- Exercise
- Next Lesson
  - The next lesson in the course is:  
**Prepare To Record**



End of Lesson

