EARLY AUTOMATION APPROACH

- By Senthilkumar Gopalakrishnan

Senthilkumar.Gopalakrishnan@cognizant.com

Abstract: 'Early Automation' approach involves automation team in the early phase of testing lifecycle. Automation scripting can commence in parallel to the system development. This method enables automation to begin at early stage of the testing lifecycle to support 'AGILE' or Iterative based projects. Parallel automation begins with understanding wireframes/proto type/use cases//Design Documents of application and analyzes the business flows. Optional sessions are conducted to touch base with the development team on need basis to sync with the development. This is followed by building object repository, creating business scenarios and Test scripts of application. The created scripts are executed in intermediate builds. If all the test scripts pass, these are added to the regression suite. If the script fails, the script is fixed and rerun. This process is continued until all the test scripts are passed and added to the regression suite

Table of Contents

1. Introduction

Automation in itself is seen as an effort/cost saving strategy in testing. How about starting the automation early in the testing life cycle! Doesn't it sound interesting?

This white paper will address the approach involved in early automation, benefits & the pre requisite required for achieving it.

2. Need for Early automation

Conventional automation advocated automation when the build is stable. Automation scripting is done in N-1 Iterations when test cases are developed and executed at least once. Industry dynamics had changed and the focus is more on cost savings and faster time to market. Automation approach has to keep pace with the emerging market trends. Many of the CEO's & Sr.Management are vying for increased coverage in automation and starting the automation when the application code is not developed. Listed below are some customer concerns:

- Why do I have to spend more cost on testing the application manually for first cycle when the decision is to automate as much as possible for regression?

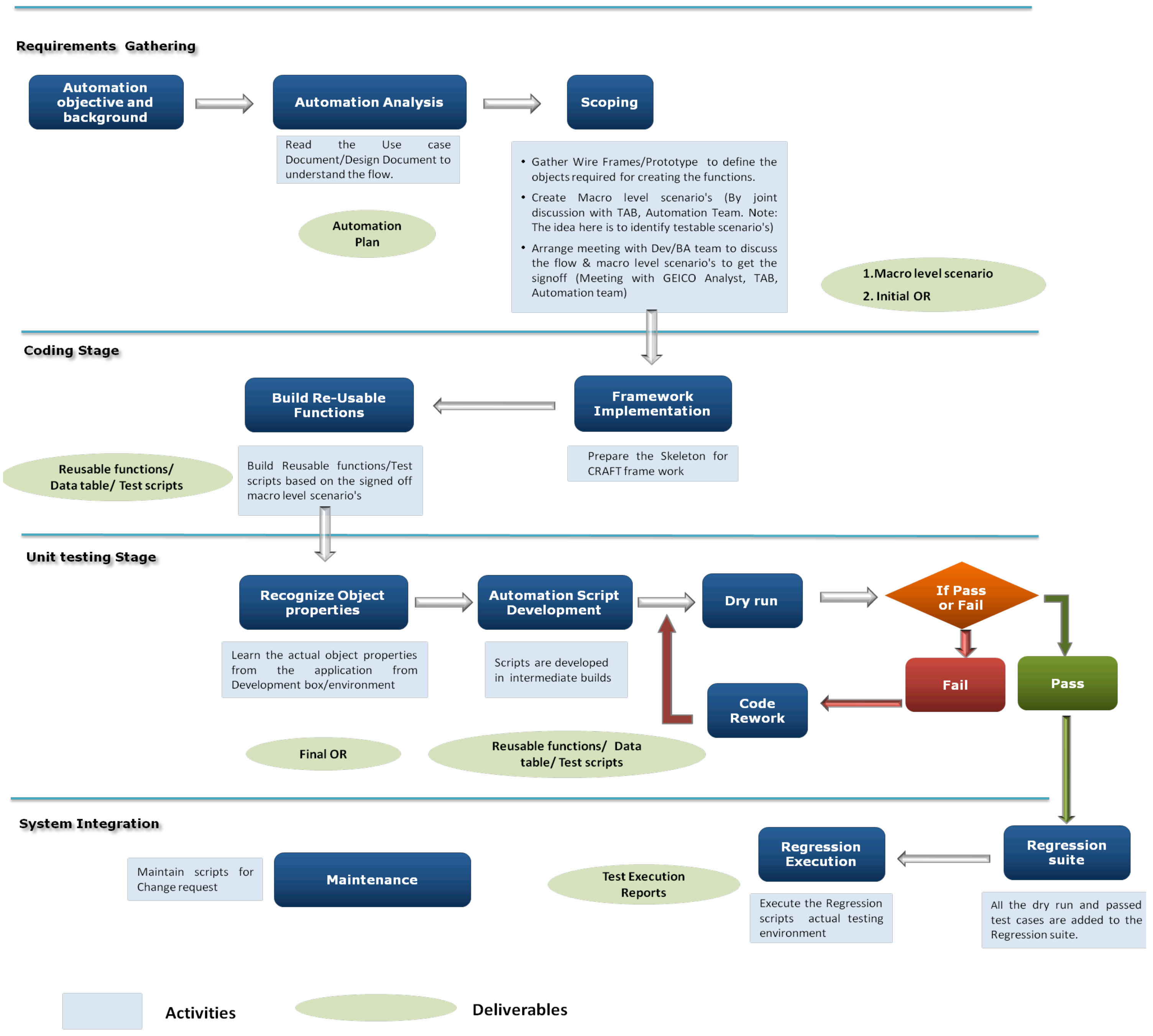
- Why can't automation start in parallel with the code development?

This document talks about picking clues from the development artifacts to start the automation early in the application life cycle.

3. Phase wise activity in Early automation

The below diagrams (Figure 1) depicts the phase wise activities involved of Early Automation approach.

Figure 1


Requirements Gathering
Automation objective and background
Automation Analysis
Scoping
Read the Use case Document/Design Document to understand the flow.
Automation Plan
Gather Wire Frames/Prototype to define the objects required for creating the functions.
Create Macro level scenario's (By joint discussion with TAB, Automation Team. Note: The idea here is to identify testable scenario's)
Arrange meeting with Dev/BA team to discuss the flow & macro level scenario's to get the signoff (Meeting with GEICO Analyst, TAB, Automation team)
1.Macro level scenario
2. Initial OR
Coding Stage
Build Re-Usable Functions
Framework Implementation
Reusable functions/ Data table/ Test scripts
Build Reusable functions/Test scripts based on the signed off macro level scenario's
Prepare the Skeleton for CRAFT frame work
Unit testing Stage
Recognize Object properties
Automation Script Development
Dry run
If Pass or Fail
Fail
Pass
Code Rework
Learn the actual object properties from the application from Development box/environment
Scripts are developed in intermediate builds
Final OR
Reusable functions/ Data table/ Test scripts
System Integration
Maintain scripts for Change request
Maintenance
Test Execution Reports
Regression Execution
Regression suite
Execute the Regression scripts actual testing environment
All the dry run and passed test cases are added to the Regression suite.
Activities
Deliverables

Requirements Phase

This is a phase where the application requirements are defined as per the business needs. Requirements document serves as a value input to the automation testers. Automation testers can pick up functional testable requirements referring to the requirement documents.

Planning Phase

This is a phase where the Use Case documents, Design documents, Wire Frames/Proto types are developed. Automation scripting can start when the application functionality, navigation flow & screen layouts are available. We can take a look at how automation can start early by picking up clues for these documents.

- ***Well Defined Use Case Document:*** Use Cases are a software modeling technique that helps developers determine which features to implement and how to gracefully resolve errors. It is very essential for the automation testers to understand the application functionality before starting on the script design. The Use Case document along with Requirement documents can help to understand the Business Functionality being implemented.

- ***Design Documents***: **A Software Design Document** (**SDD**) is a written description of a software product, that a software designer writes in order to give a software development team an overall guidance of the architecture of the software project. An SDD usually accompanies an architecture diagram with pointers to detailed feature specifications of smaller pieces of the design. This document can be used by automation testers in script designing. They can get to know the navigational flow of the functionality implemented.

- ***Wire Frame/Prototype***: A prototype is an early sample or model built to test a concept or process or to act as a thing to be replicated or learned from. The automation developers can refer to this to define the object wither in object repositories or define descriptive programming in the actual functions are scripts. Even though the exact object properties might not be available in the proto types or wire frames, this can act as a reference to define the dummy objects required for scripting.

- ***Macro Level scenario:*** Macro scenarios are miniature test cases covering the business rules & the functionalities. This can be developed by the Functional testers. The automation testers can be involved in the preparation of the macro scenarios.
- ***Optional session with Business Analyst/Development:*** In most of the projects, a regular stand up meeting is held to discuss the project health. This can be used as touch points

for the automation testers to sign off for the macro level scenarios that have been developed covering all the requirements.

The automation testers can pick up the objects, functionalities & navigation flows from the above and start creating the object repositories, reusable functions, data tables & then combine them to develop the test scripts covering the macro level scenario. A hybrid framework (combination of data driven & keyword driven framework) is recommended for this approach to keep the maintenance at minimal level.

Unit Testing Stage

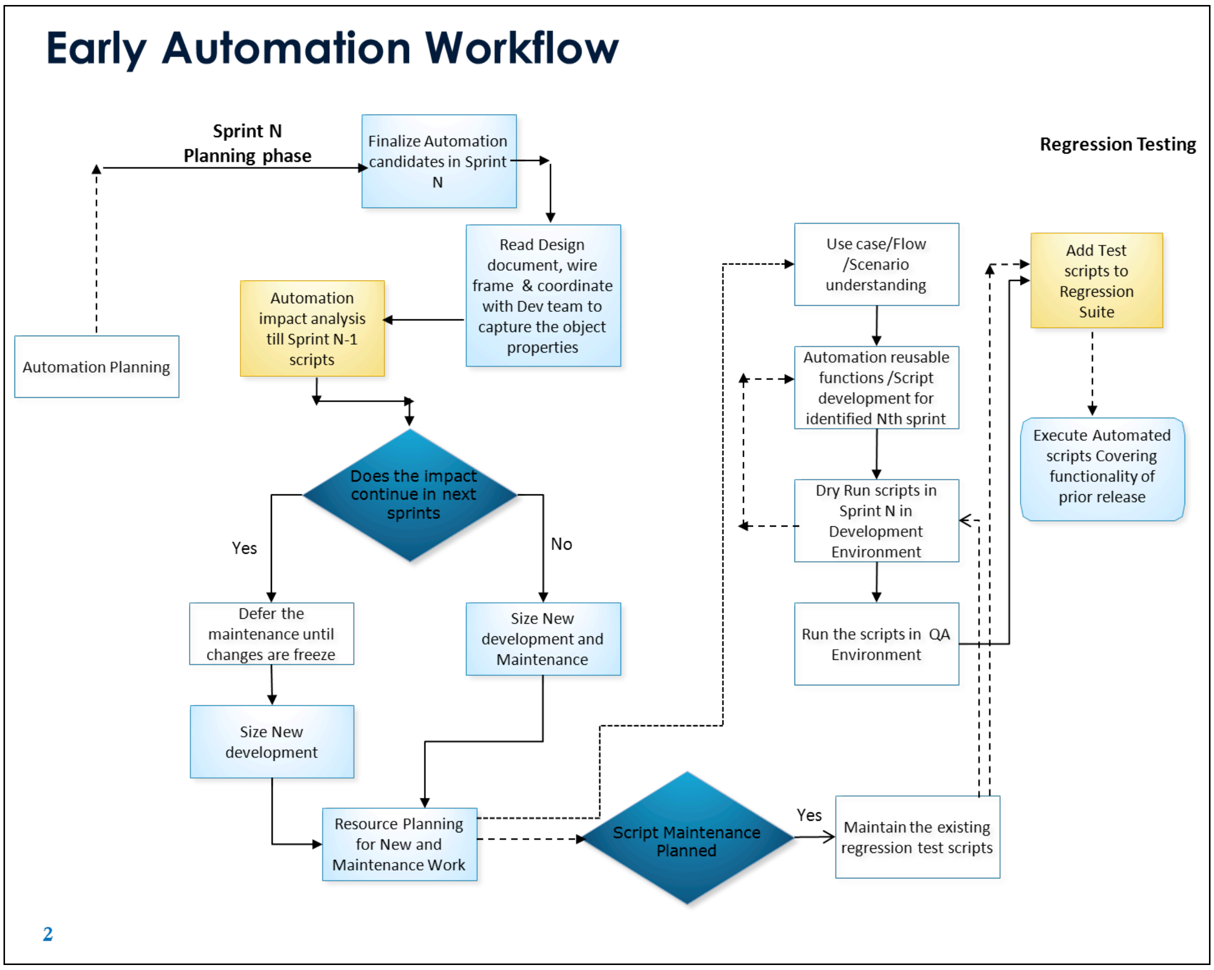
This is a stage when the actual application is available in partial or fully developed form where the developers/testers conduct the unit testing. The actual object properties would be available and can be captured at this stage. The developed scripts are executed in this stage either directly in the developer's box or in an environment where this build is available. If all the test scripts pass, these are added to the regression suite. If the script fails, the script is fixed and rerun. This process is continued until all the test scripts are passed and added to the regression suite. Utmost care should be taken when the scripts are modified, that the script developed verifies the intended functionalities rather than taking the application behavior in to consideration. Any defect found at this stage can be logged and tracked to closure, thereby helping in early identification & closure of defects.

System testing stage

This is the stage when the actual build is pushed to the System testing environment. The regression script is executed during this stage and failure report and Test execution summary report are presented.

4. Early Automation work flow

The below diagram depicts the flow in Agile environment


Early Automation Workflow
Sprint N Planning phase
Finalize Automation candidates in Sprint N
Regression Testing
Read Design document, wire frame & coordinate with Dev team to capture the object properties
Automation impact analysis till Sprint N-1 scripts
Automation Planning
Use case/Flow /Scenario understanding
Add Test scripts to Regression Suite
Automation reusable functions /Script development for identified Nth sprint
Does the impact continue in next sprints
Yes
No
Dry Run scripts in Sprint N in Development Environment
Execute Automated scripts Covering functionality of prior release
Defer the maintenance until changes are freeze
Size New development and Maintenance
Run the scripts in QA Environment
Size New development
Resource Planning for New and Maintenance Work
Script Maintenance Planned
Yes
Maintain the existing regression test scripts
2

Early Automation – Entry and Exit Criteria

Category		Description
Automation Estimation	Approach	▪ Based on the Requirements /use case document automation complexity will be arrived and will be converted into Test script points (TSP). The effort will be estimated for the TSP.
	Entry and Exit Criteria	Entry: ▪ Business Requirements document should be available for the Automation team ▪ Proto type/wireframes should be available ▪ Use case documents should be made available Exit: ▪ Completed effort estimation for Automation ▪ Resource Loading plan
	Accelerators	▪ More resource pools working to support QA activities and to increase automation levels
	Assumptions	▪ 6 Business Rules will be combined into one Manual scenario ▪ Actual estimates may vary depending on the comprehensive test case analysis, feasibility analysis and identification of complexity factors across all applications ▪ Any changes in application user interface or test case flow from the baseline scenario will result in automation rework and is not included as part of this estimate

Automation Design	Approach	▪ Understand the requirements/ user cases by having detailed session with BA & Developers ▪ Identify feasible scenario's to be automated ▪ Understand the Test Scenario ▪ Analyze the test data & business scenarios ▪ Extract Object properties from Wire Frames to create object properties or descriptive programming ▪ Identify reusable components ▪ Creating reusable functions ▪ Creating scripts based on the test scenario document and screen shots ▪ The created scripts are executed in intermediate builds. If all the test case passed, these are added to the regression suite. If the script fails, the script is fixed and rerun. This process is continue until all the test cases are passed and added to the regression suite
	Entry and Exit Criteria	Entry: ▪ Detailed KT on each use cases to be provided to the Automation team ▪ Test data and precondition data will be made available Exit: ▪ Automated Test script completed for the scenarios that is feasible. ▪ Sign off for Test script from Customer.
	Accelerators	▪ Creating reusable functions from Wireframes/Prototypes/Use cases ▪ Creating scripts for failed test scenarios till the failed step ▪ Parallel Automation design with Code development.

	Assumptions	▪ The test environment will be provided, owned and maintained by customer ▪ Customer will facilitate a detailed KT session on the different modules in the application to the Cognizant Automation Team which are in scope for automation ▪ The number of test cases that can be automated in each of the application in scope for regression automation depends on the outcome of the feasibility analysis phase. ▪ Any updates in the test scenarios / screens will be communicated to automation team through proper channel and the estimates will be revisited if there is a huge update.
Automation Execution	Approach	▪ Completely Automated test scripts list for each sprint will be pulled out. ▪ Automation Execution Plan will be prepared based on the scripts that are available for execution ▪ Systems will be allocated for execution and the execution will start. ▪ Detailed Summary report with results will be shared with the customer
	Entry and Exit Criteria	Entry: ▪ Environment Availability and dates of execution ▪ No Show stopper defects ▪ List of known defects Exit: ▪ Execution completion. ▪ Defect logging ▪ Detailed Automation Test Summary Report

	Accelerators	▪ Execution in multiple systems - Reduce ~30% of the execution time ▪ In case of unattended execution, the resource can be utilized on Automation design ▪ Batch execution from Run Manager
	Assumptions	▪ Application changes will be very minimal ▪ Test data and precondition data will be available

General Recommendation:

- Customer will facilitate a detailed KT session on the different modules in the application to the Cognizant Automation Team which are in scope for automation.
- Cognizant assumes that a BA/ SME from its side will be available for providing the necessary inputs, clarifications and documents to the Cognizant team.
- To consider Formal functional specification or requirements document that has testable requirements
- Tech spec detailing the areas that are impacted
- Test scenarios along with detailed steps and validations for automation
- Development team should following proper naming conventions which is mutually agreeable
- Screen layouts and design for the proposed requirements
- Dedicated environment for Automation where the new screens are made available as it was developed, so that the automation team can start creating the reusable functions as early as possible i.e. the develop environment and testing environment should have the same build updated in it.
- Master / Base data to be set up and to be loaded in all new builds / environment so that most of the data for Automation remains same and thus will not increase execution effort
- Communication from the development team to the automation team regarding the changes being made to the already delivered screens/modules
- Development implementation schedule by screens / functionalities along with availability for the automation team to read the screen objects and fields
- Execution of test scripts in multiple machines and unattended execution will save more effort

5. Benefits

As all best practices have few benefits, this approach is no exception. Few of the major benefits are listed below

- Free up Functional testing resources and enable them to devote effort for critical tasks
- **Reduction in overall testing effort:** To give one example on how this approach can result in effort savings, you can take a huge project. Conventional approach involves writing manual test cases, executing them in one cycle before they are converted to automation tests. In a long run after the breakeven point, the ROI will be higher in early automation approach than that of conventional automation approaches. The below sampling will help in understanding the effort & cost savings.

	Conventional Approach			
	# of Tests	**Test case creation time in hrs**	**Execution time in hrs**	**Total**
Manual	3000	250	375	625
		Test case productivity =12 Test case s/hr Test case Execution=8 Test case execution/hr		
	Early Automation Approach			
	# of Tests	**Test script creation time in hrs**	**Execution time in hrs**	**Total**
Manual	1200	100	150	250
				250
		Average automation % Considered =60%		
Manual Effort Saved through early automation				375
Average Billing Rate				$55
Cost Savings/Cycle				$20,625
# of Cycles				6
Overall Manual Effort Saved				**$123,750**

- **Early identification of Defects**: As explained in the above mentioned unit testing phase, this approach can help in early identification & mitigation of defects. Also freeing up Function testing resource can help this cause.

6. Conclusion

Having talked about the benefits of "Early automation approach", there needs to be a high degree of discipline maintained to reap better results which can otherwise lead to more maintenance effort. This involves standards to be followed by developers in documenting the use cases, design documents & prototypes. Automation testers & Test Analyst have to be cautious while designing the macro level scenarios. The co-ordination between the Developments, BA & QA teams had to be in a closed loop in order to communicate the changes in any of the artifacts.

7. Biography

Senthilkumar Gopalakrishnan is a Sr. Project Manager in Cognizant Technology Solutions with 10 years of industry experience in automation. He has been involved in providing automation solutions for top tier clients in various domains like Insurance, BFS, Healthcare and Life science. He also heads the Automation CoE operations for North America region.

Thank You