Mentorship Program

Tuesday, January 17, 2017

12:41 AM

Priyanka Ghose

Thursday, December 22, 2016

6:00 PM

* + How to explain person x his/her priorities?
  + Person X feels Y is given more importance and at the same time X expects to get reward for the work. How to handle this situation?

Prathima Gajarala

Thursday, December 22, 2016

6:00 PM

Team Leaves 2017

Friday, May 05, 2017

7:04 PM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Morning Shift | General Shift | Flexi Time | Leave | Others |
| Deekshith |  |  |  | ~~29th, 30th June and 3rd July~~ |  |
| Tribhuvan |  |  |  | ~~23rd June~~ |  |
| Sudha |  |  |  | ~~16th June~~  25th August |  |
| Sruthi |  |  |  | ~~17th May 2017~~  ~~13-15 June 2017~~  25th August |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Team Efficiency

Monday, September 11, 2017

2:18 PM

|  |  |  |
| --- | --- | --- |
| **Name** | **# Test Cases/Month** | **# Test Cases Updated/Month** |
| Sudha Chakinala | 3.45 |  |
| Sruthi Kura |  |  |
| Durga Prasad Vemula | 3.37 | 26.28 |
| Usha Sai Masabattula | 2.41 | 23.76 |
| Shireesha Gaddam | 3.03 | 21.3 |
| Priyanka Kudikala | 2.07 | 10.9 |
| Charuvaka Thalapelli | 2.45 | 17 |
| Santhoshi Chepuru | 2.21 | 23.2 |
| Ranvijay | 4.5 | 7.6 |
| Komal | 2 | 6.66 |
| Kiranmai | 1 | 11.16 |
| Vasavi | 0 | 20 |

Project Priorities

Friday, May 12, 2017

11:11 PM

**Plan to Move on GIT:**

**Set up Systems to fix lint errors:** 15th May, Monday (Bhupender)

**Prepare Training material:** 22nd May (Deekshith)

**Give Training:** 24th May (Deekshith, Tribhuvan and Bhupender for backup)

**Releasing PRU project to GIT (new Structure):** 26th May (Tribhuvan) - 1st Trial on 19th May from tellusdev (if tellusdev is ready)

**Regression Branch Testing:** Ranvijay, Sruthi

**Other Prebuild Testing:** Komal, VNKiranmai (Charuvaka as a backup)

**GIT:** Vasavi and Sudha

**SoapUI:** Priyanka

Moving QA repository into PA3 repo

Tuesday, August 29, 2017

2:12 AM

Objective : Move QA Repo to Pa 3 App repo

When we move QA repo (as sub-repository) inside PA3 repo,

we would have two package.json, one for PA3 repo and another is inside qa-repo (which have qa dependencies listed) There will be a configuration file which would drive e2e scripts.

Approach #1 **[Agreed to this approach]**

If we decide to use package.json file from PA3 repo, one has to ensure that qa-dependencies version are up to date with the one found inside qa sub-repo. And as we would bump up the version of dependencies based on the build releases there need to be frequent updates made to PA3 repo's package.json.

Approach #2

if we switch to qa sub-repo we don't have to worry about synching PA3's package.json with qa's package.json as we can do npm install on qa's package.json which will take care of installing test dependencies.

Now in order to execute e2e test we have to consider the following two cases:

1. Drive it using the configuration file maintained at PA3 repo level which **would either call some set of test cases or bvt test case or whole automation suit**e.
2. Package.json file maintained at PA3 repo level should include qa-test dependencies.

However, we have few questions:

1. There are some lint Standard which may have been suppressed for QA Scripts which would cause error. Is it possible to use qa's lint setting file for running **gulp lint** command (which verifies all lint issues for the given project)?

**Ans**. QA script will be excluded from engineering lint standards.

1. QA might make many commits and releases the same which would increase the count of commit for PA3 repo. Is it acceptable to engineering if have many commits in qa repo?  
   **Ans**.: Jenkins should cd into qa directory and we would still be able to do **gulp release**.
2. Is it possible to exclude qa sub-repo while PA3 releases its next version as we might have nothing to release at that point in time as all changes from qa side would have already been released and tested?  
   **Ans**.: It'll be part of PA3 repo release.
3. Are gulp commands acceptable to Engineering ?  
   **Ans**.: Yes
4. What happens when Engineering is going thru code freeze and QA have to make script changes ?  
   **Ans**.: We can raise MR and accept it.
5. Does engineering want to run full suite or just BVT?

**Ans**.: Full suite.

1. How do engineers create their prebuild and allow us to test the same? Do you host your changes to some web-server where application picks the code when we hit that prebuild?

**Questions for Analysts to setup workflow after moving QA repo into Engineering repo**

* Consolidate test cases into feature based test plans and have the mapping for the same.
  + Engineer should be able to identify the suite of test cases which test the particular functionality of the application.

**Question:** Is our existing inventory aligned this way? If yes, do engineer know which test plans are testing which part of application?

* After the changes made to test script by Engineers as per the enhancement to application:
  + Merge request will be filed by Engineers and Analyst will be involved in the same.
  + Test cases must be updated accordingly and both the parties should review the same.

**Question:** How are we updating existing test case which is already in production?

* Do you maintain staging version of test case? If yes, when you move the changes to production version of the test case do you retain the same test step ids which already existed in production version of test case?
* If new test steps are added to existing test case do you move the steps to production with same test step ids or does it change? (Here, I am assuming new steps are added to staging version of test case).
* Do you maintain staging test case with changes which are supposed to be released to production sometime later, say 2-3 weeks?

Workflow after moving QA repository into PA3 repo

Sunday, September 24, 2017

3:31 PM

* 1. QA Automated scripts will be placed under PA3 repository under the folder (TBD).
     1. Dependencies of QA repository will be added to PA3's package.json.
     2. Package.json will be remove from QA repository.
     3. Lint exceptions will be made for qa sub-directrory.
     4. Configuration file, which will have mapping to automated test cases, will be placed under PA3 repo.
     5. Engineers will be able to execute these test cases by running just this configuration file.
  2. During application development phase, automated scripts (full automation inventory) will be executed.
  3. If scripts fail and it is due to changes/enhancement in the application, engineers will update all the affected scripts and will raise Merge Request for scripter to review the same.
     1. Analyst and Scripter review the changes and may request for changes to be made.
     2. Once changes are confirmed, Analysts will update manual test case.
  4. If script failure is due to logical error in script then gitlab issue will be raised and scripter will fix the issues reported.
  5. Tellus framework or tools related issues, which might come up during execution of scripts locally, will be handled by a SPOC from scripter's team.
  6. Script changes/enhancement would be released along with promotion of engineering code.
  7. For making the changes to script after code is promoted, scripter can raise Merge Request to sneak the same either to Devel, QA or Live code base.
  8. During the code freeze also scripter can raise Merge Request if there are any changes to be merged for automated scripts.
  9. Jenkins configuration has to be updated to release qa sub-directory as node module as tellus expects qa repository as node\_module to execute test scripts.
     1. When engineers promote the code, either their Jenkins configuration would call our Jenkins configuration or qa's Jenkins configuration would call theirs in order to release qa scripts into artifactory.
  10. Jenkins configuration will be executed whenever MR is accepted and new version of qa scripts would be released to artifactory.
  11. QA should still be able to release the changes/script bug fixes/refactor/addition of new scripts to artifactory through explicitly running jenkins **build now** option.
      1. This should still happen through Merge Requests.

**Questions to be answered**

* 1. How to get metrics on bug identified through automated scripts? Can we have engineers create label/tag and add it along when fix for the issue found through automated scripts is submitted?

**Questions for Analysts**

* 1. Consolidate test cases into feature based test plans and have the mapping for the same.
     1. Engineer should be able to identify the suite of test cases which test the particular functionality of the application.

**Question:** Is our existing inventory aligned this way? If yes, do engineer know which test plans are testing which part of application?

**Ans.:** Test plans are already aligned to dedicated features of the application.

* 1. Prathima's team will provide the cheat sheet for engineers, which will have mapping of test plans to its features.

**Question:** How are we updating existing test case which is already in production?

* 1. Do you maintain staging version of test case? If yes, when you move the changes to production version of the test case do you retain the same test step ids which already existed in production version of test case?  
     Ans.: Staging version is maintained until changes are pushed. Script update is raised based on staging version of test case. Once new changes are moved to production version of test case , script changes will also be moved to production (here step ids would remain same).

**NOTE:** When script update request is raised it is referred from staging version of test case unless the changes are not already seen in Live. If the changes are present till live, production version of test case is updated.

* 1. If new test steps are added to existing test case do you move the steps to production with same test step ids or does it change? (Here, I am assuming new steps are added to staging version of test case).

**Ans.:** Same step ids are maintained in production version too.

* 1. Do you maintain staging test case with changes which are supposed to be released to production sometime later, say 2-3 weeks?

**Ans.:** Staging version is maintained and script update request is raised. But, changes are moved to production only when those enhancements are moved to production.

**TO Dos**

* 1. Weekly meeting with engineers, analysts and scripters to discuss concerns/issues during this process.
  2. Identify scripter who'll be SPOC for reviewing Merge Requests.
  3. Documentation on scripting standards to be followed.
  4. Tellus team adding support for testing forked projects (Testing in progress).

**Solutions from Harsha for releasing QA modules**

PA3's package.json will be updated to run gulp tasks for QA.

Workflow for updating QA automated scripts

Tuesday, September 26, 2017

12:31 AM

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Responsible** | **Task definition** | **Questions** |
| Moving QA code inside PA3 repo | Automation Team | Scripter will move automated scripts inside PA3 repo |  |
| Update **package.json** of PA3 repo | Engineering | Engineer should update their **package.json** to include qa dependencies. |  |
| Lint exceptions for QA | Engineering | Exclude qa sub-directory from lint rules of PA3 project |  |
| Protractor configuration file | Automation Team | Map automated test cases into configuration file before placing the same into PA3 repo |  |
| Execute Automation Scripts | Engineering | Engineers can run automated suite during the development phase or before handing over prebuild for Regression Testing. |  |
| Update Scripts due to Application changes/enhancements | Engineering | If scripts would fail due to changes in application code, engineers would update all the affected scripts and raise Merge Request for scripters and analysts to review. |  |
| Update Manual Test Cases | Analysts | Once changes are confirmed, Analysts will make changes to manual test case to be in synch with application enhancements. |  |
| Update scripts to fix logical errors | Automation Team | If script failure is due to logical error, gitlab issue will be raised by engineers for scripter to fix the same.  Once the script fixes are done, the scripter will open MR to merge the changes into main qa sub-directory. |  |
| Framework or tool related issues | Automation Team | If engineers come across issues related to either tellus framework or automation tools, SPOC from automation team will follow up with respective teams to clear out the same. |  |
| **Releasing QA Changes** | **Engineering** | **Along with release of application code by engineers, qa code will also be released as new version.** | **As tellus needs qa project as npm module, how qa sub-directory would be released to artifactory when application code is promoted?** |
| **Sneaks for script fixes after promotion** | **Automation Team, Engineering** | **If scripter makes fixes to script after new version is released, they should raise MR.**  **Engineers will review the same and accept the same.** | **How the script changes would be released to artifactory? Currently, we release the changes as patch fixes.** |
| **Merging script fixes during code freeze** | **Automation Team, Engineering** | **During code freeze time, scripter can still submit the MR which will be accepted by engineers after the review.** | **After accepting MR during code freeze, will it be released to artifactory?** |
| **Updating Jenkins configuration** | **Automation Team** | **In order to release qa sub-directory, Jenkins configuration has to be updated such that it’ll cd into qa sub-directory and run gulp release process.** | **Currently, we have Jenkins configuration setup which uses webhook of qa repository to detect the push event and then it'll publish the changes to artifactory. What changes are required to jenkins configuration when qa project will reside in pa3 project?**    **Also, we use gulp as a release workflow and PA3 team uses grunt. Are there any plans to switch over to gulp? Or do we have similar process in grunt to publish npm modules to artifactory (not sure if this question is for developer services team)?** |
| Release on accepting MR | Automation Team | Once merge request is accepted, Jenkins job should be triggered in order to release qa sub-directory as npm module into artifactory. |  |
| Script bug fixes/Code refactor/Addition of new script | Automation Team | Either script is updated for fixes or it is refactored or new script is added to inventory QA should be able to trigger Jenkins build to release the changes into artifactory.  **NOTE:** Scripter should still raise MR in order to accept these changes. |  |

**TO Dos**

1. Weekly meeting with engineers, analysts, and scripters to discuss concerns/issues during this process.
2. Identify scripter who'll be SPOC for reviewing Merge Requests.
3. Documentation on scripting standards to be followed.
4. Tellus team adding support for testing forked projects (Testing in progress).

Machine generated alternative text:
Yes (Purely script issue) 
Issue in gitlab is 
opened and Scripter 
Will look into it 
Scripter fix the issue 
and submit the same 
to merge into QA sub- 
directory 
Changes 
suggested 
Script Failed 
s this ascrip 
• No (Application Changes) 
QA repository residing in 
PA3 repository 
New changes made to 
No action 
application 
required 
Analyst Review 
alysts are fine With updates an 
make changes to test case 
accordingly? 
Script can be merged 
to QA sub-directory 
Analyst suggest changes as to 
how testing of new functionality 
should be written 
Engineer Will update all 
the affected script and 
submit a merge request 
Scripter and Analysts 
work together to reviews 
the code changes 
Scripter Review 
QA code 
standards/best 
practices are 
followed? 
Scripter suggests changes to be made 
anges are reviewe 
and accepted? 
Changes are merged 
to qa-subdlrectory 

Notes

Thursday, October 12, 2017

7:41 PM

**Points to discuss with Rajul**

1. JSDoc for generating documentation for Protractor.
   * We need to align our page objects to use concept of classes which comes with ES6.
   * Use ES6 features and standards for script development.
2. Code coverage tool which can give report on coverage of application code.
3. Work on documentation of code standards.
4. Speed Sprint Tasks selected for Analytics.
5. Meetings with Engineering teams.
6. Merging test and page object repos.

Book Reading - How Google Tests Software

Tuesday, October 17, 2017

8:04 PM

* 1. Software testing is part of a centralized organization called Engineering Productivity that spans the developer and tester tool chain, release engineering, and testing from the unit level all the way to exploratory testing.
  2. Quality can't be tested in; It is equally evident that without testing it is impossible to develop anything of quality.
     1. Stop treating development and test as separate disciplines.
     2. Quality is not equal to test. Quality is achieved by putting development and testing into a blender and mixing them until one is indistinguishable from the other.
  3. Concentrate more on quality of product rather than releasing more features to client.
  4. Google testers are willing to try anything once but are quick to abandon techniques that do not prove useful.
  5. The absence of plenty forces us to get good at prioritizing, or as Larry Page puts it: “Scarcity brings clarity.”
     1. Scarcity also makes testing resources highly valued. The first piece of advice I give people when they ask for the keys to our success: Don’t hire too many testers.
  6. Everyone who writes code at Google is a tester.
     1. The burden of quality is on the shoulders of those writing the code.
     2. If you are an engineer, you are a tester.
     3. Developer own quality.
  7. If a product breaks in the field, the first point of escalation is the developer who created the problem, not the tester who didn’t catch it.
     1. Quality is more an act of prevention than it is detection.
     2. Quality is a development issue, not a testing issue.
  8. Testing must be an unavoidable aspect of development, and the marriage of development and testing is where quality is achieved.
     1. Testers are there to make developers more productive.
  9. SETs are partners in the SWE codebase, but are more concerned with increasing quality and test coverage than adding new features or increasing performance. SETs write code that allows SWEs to test their features.
  10. The Google TE is a mix of technical skills that developers respect and a user facing focus that keeps developers in check.
      1. Puts testing on behalf of the user first and developers second.
      2. TEs are product experts, quality advisers, and analyzers of risk.
  11. SWEs own features and the quality of those features in isolation. They are responsible for fault-tolerant designs, failure recovery, TDD, unit tests, and working with the SET to write tests that exercise the code for their features.
  12. TEs can turn to the primary task of ensuring that the software runs common user scenarios, meets performance expectations, is secure, internationalized, accessible, and so on.
  13. Google’s reporting structure is divided into what we call Focus Areas.
  14. Test exists in a separate and horizontal (across the product FAs) Focus Area called Engineering Productivity.
  15. If a development team wants us to take any shortcuts related to testing, these must be negotiated in advance and we can always decide to say no.
  16. Testers are assigned by Engineering Productivity leads who act strategically based on the priority, complexity, and needs of the product team.
  17. The on-loan status of testers also facilitates movement of SETs and TEs from project to project, which not only keeps them fresh and engaged, but also ensures that good ideas move rapidly around the company.
  18. It is generally accepted that 18 months on a product is enough for a tester and that after that time, he or she can (but doesn’t have to) leave without repercussion to another team.
  19. Encourage culture to build small and test small and move ahead.
  20. Divide testing into the silos : Small, Medium and Large tests.

Training to Vinil

Thursday, July 27, 2017

7:02 PM

* 1. Different product groups.
  2. Products we currently deal.
     1. Product wise allocation of resources.
     2. Tools used for automation.
  3. What is QA Attributes and why it is required?
  4. Thief helper libraries and why it is required.
  5. Automation feasibility check.
  6. Project structure
     1. Protractor
     2. Ruby
  7. Performance machine setup
  8. Prebuild testing and how we track the same
     1. PA3 regression branch
  9. Weekly Team Updates and different information collected.
  10. RPD workflows For
      1. Bug found during script development
      2. Script failed but changes are expected
      3. Script failed and changes are not expected
      4. Script failed but no changes were made from application side
  11. Call It RPDs
  12. RPD Tags we use and its purpose.
  13. RPD views and its purpose
  14. Tellus framework
      1. JEP
      2. Calendar
      3. Progress Page
      4. Tellus Test Manager
      5. Automation Status
      6. Tellus List Builder
      7. Tellus Result Page
      8. Scheduling jobs
      9. CRON jobs
      10. Job Scheduler and priorities of incoming job.
      11. FPE Vs SGE Vs SoapUI
      12. Launching automation from SWP and how status are updated

Requirements for Rajul's Replacement

Monday, December 18, 2017

1:20 PM

* + Should be able to suggest and implement the best processes for QA functioning.
  + Collaborate with SBU heads and should explain the importance of aligning SBU QA roles to global standards.
  + You should be responsible for scaling the home grown framework (Tellus) to cater the needs of increasing automated testing.
  + Should be proficient in understanding the functioning of Manual and Automation roles such that he can drive the efforts to transform QA at FactSet.
  + Identify individual potential and fit them in a role which can reap maximum benefit and will create win-win situation for both employee and FactSet.
  + You should be responsible for collecting metrics form SBU QA heads and analyze the effectiveness of QA.
  + Build the platform to enable QA working closely with Engineering and PDs and understand each other processes.
  + You should create opportunity to enhance the skillset of employee by designing training curriculum which will help QA to grow.
  + Responsible to build relationship with various teams such as Engineering , PDs, Infrastructure etc., so that QA challenges can be addressed timely.
  + Should have strong knowledge on QA process, Automation Tools and technologies.
  + Should make decisions after thorough analysis of given scenario.
  + Should be part of hiring panel to judge technical aspect of job seeker.
  + Responsible to drive and implement testing pyramid.

Job Metrics Needed

Tuesday, January 16, 2018

2:07 PM

* + No. of Prebuilds tested in a month over month.
    - Number of scripts executed for each prebuild?
    - Number of Bugs Found with prebuild testing?
  + No of Script Intermittent issues found?
    - Any measures taken to reduce the same?

Objective of Integrating QA repo into Application repo

Friday, March 09, 2018

11:09 PM

**Objective:** With the integration of QA repo into application repo in GitHub we want to **enable engineers to be able to consume automated scripts** as and when required during development process which in turn help to **reduce the feedback time on the code quality** written by an engineer. Following are the benefits we foresee with this strategic move:

* **Instant feedback** on code changes made by an engineer
* Enable engineers to **run full e2e tests and identify code flaws even before build creation**
* **Reduces testing cycle time** once the build is out for testing
* **Changes are released to market much faster**
* Developers being able to **identify automation challenges** and **develop automation friendly application**
* **Eliminate the need to release QA projects to artifactory**
* **Reduce maintenance efforts** of automated scripts
* **High collaboration b/w QA and Engineering** such that both teams learn the best practices from each other's processes

Monthly Highlights from CoE - February 2018

Monday, March 12, 2018

12:10 PM

We had some good sessions with participation from all leads from given SBU where we discussed on various topics such as scripting challenges, new ideas and ways to speed up automation process.

The report include the following sections:

1. Scripting Challenges
2. New ideas from Monthly knowledge Sharing sessions
3. Strategic Projects
4. Preview for March

**Scripting challenges:**

<<KnowledgeSharing.xlsx>>

Machine generated alternative text:
Automation Tool 
Test Complete 
Selenium-Ruby 
Selenium-Ruby, 
Protractor 
Protractor 
Problem Statement 
Fetching passcode or other content from email body or 
subject 
Job execution hangs on unexpected window 
Get chart references and retrieve data from the same 
unable to access elements present in Shadow Root 
tree 
Resize native window dialogs 
Capture Network Logs and Memory usage information 
during automated script execution 
Moving workstation dependent project from Perforce 
to GitHub 
Description 
There are scenarios when you need to fetch some data from email subject 
or body which will be used to perform actions on other components 
based on the data received in email. 
To achieve this we need a way to connect to outlook and fetch the data 
required. 
Scripters generally ensure that they are handling all the negative 
scenarios gracefully which should not interrupt the execution of job. 
However, it is not possible to know all negative scenarios which leads to 
unexpected windows. 
It is observed many times that job execution hangs on unexpected 
window which stops execution of next steps. 
For web-application we have nice support to automate charts using 
FDSChartJS but the same is missing for PC side application. Currently, we 
are not able to automate charts for PC side applications. 
There is a concept in JavaScript to limit the scope of HTML which is 
achieved through adding shadow root element and it is not possible to 
get element references found under this root using regular XPATHs. 
There are test cases where you are asked to resize the window and these 
windows are not HTML based. Thus, we use AutolT to perform actions on 
any non-html components. 
Which SBU faced this issue? What is the solution? 
We have to create outlook object and then use inbuilt utilities with Test 
Complete to get data from email 
SmartBear has provided a documentation on how to handle 
unexpected/unhandled windows. Here are the links: 
https://support.smartbear.com/articles/testcomplete/handling- 
unexpected-windows/ 
https://support.smartbear.com/testcomplete/docs/testing- 
with/running/handling-errors/unexpected-windows.html 
Smartbear suggest to use OCR technology which can read data from 
images but how much it can read data from charts is still a question 
One has to use css as locator strategy along with the combination of 
'deep/ to access elements from shadowroot tags. 
While researching on alternative tools to AutolT we found out another 
tool named pywinauto which can automate native windows. However, 
one has to know Python scripting language in order to utilize the same. 
Vasu found out that there are list of other tools some of which come 
default with windows OS which can be used to automate native windows 
On doing some research we found that community suggests 3 different 
appraoch to achieve this: 
1. using Firebug extension - Limitation of this is, it works only for Firefox 
browser 
2. using BrowserMobProxy class and create a local proxy such that all 
network traffic is route through this proxy - Have to check if this works 
and complications involved in implementing this 
3. use fiddler to capture network logs - Seems to be promising solution 
but it has its own issues related to memory consumption 
Priyanka M from Research SBIJ developed an executable file which can 
launch workstation. Now, we will be able to use this binary file and get 
rid of Seleniu-Ruby project dependency and hence can move remaining 
projects to GitHub. 
"of Project which has such dependecy: (need to_get detailo 
Platform 
Platform 
Platform 
Platform 
Platform 
There are times when you want to capture the network traffic and also get 
browser memory usage during automated script testing. These logs will 
Research 
help application developers to assess Performance of application and also 
help to debug the issues faster 
We still have few projects developed using Protractor and still have 
Perforce as it Source Control Manager. 
This is due to dependency on workstation. 
Research 
Currently, workstation is launch using Ruby utilities for Protractor projects 
and thus such projects rely on two differnt type of projects developed 
using different automation tools. 

**New ideas from Monthly knowledge Sharing sessions:**

1. **Introducing pair programming** to improve the quality of automated scripts
2. A white paper describing the **guidelines to initiate POCs**
3. Identify the **requirements to implement the Phase 2 of QA Transformation**

**Strategic Projects:**

1. **POC to Move Soap UI projects to GitHub**

**Objective:** Run Soap UI Projects as part of CID

**Description:** Currently, our Soap UI projects reside on Perforce and this it is not possible to run the same as part of every new build that gets triggered from CID. Thus, moving Soap UI projects on GitHub allows us to configure Jenkins to pull Soap UI projects from artifactory and execute it for every build triggered by CID.

**Benefits:**

a. We can have web-services tests run as part of CID

b. Faster download time for Soap UI project when the job is scheduled on tellus

c. Elimination of downloading unnecessary files which are not required for Soap UI project

d. Simplifies the execution of Soap UI project as one can install the project as a Ruby gem

**The process to move other Soap UI projects:** [Soap UI projects on GitHub](onenote://H:/Department/QualityAssurance/QAAutomation/TellusFramework/TellusFramework/Center%20of%20Excellence.one#Soap%20UI%20projects%20on%20GitHub&section-id={5179D27B-2C8B-4A4F-8388-11B61E798A5C}&page-id={89B52A9E-FD59-4E74-A44F-C792E5F660CE}&end)

**Current Status:**

* POC is completed.
* Full inventory of projects to be provided shortly

1. **Integration of QA branch into application repo**

**Objective:** With the integration of QA repo into application repo in GitHub we want to **enable engineers to be able to consume automated scripts** as and when required during development process which in turn help to **reduce the feedback time on the code quality** written by an engineer.

**Benefits we foresee with this strategic move:**

* **Instant feedback** on code changes made by an engineer
* Enable engineers to **run full e2e tests and identify code flaws even before build creation**
* **Reduces testing cycle time** once the build is out for testing
* **Changes are released to market much faster**
* Developers being able to **identify automation challenges** and **develop automation friendly application**
* **Eliminate the need to release QA projects to artifactory**
* **Reduce maintenance efforts** of automated scripts
* **High collaboration b/w QA and Engineering** such that both teams learn the best practices from each other's processes

**Current Status:**

POC for this integration is initiated with *PA3 application* and Vinil is leading this effort.

**Preview for March:**

Activities that will be included in March report includes:

* Reviewed automation development in PMT group and suggested ways to improve automation pace
  + Code Optimization
  + Generalized utilities
  + Adhere to Page Object model
* Documented differences b/w Soap UI Pro and Soap UI Open Source version