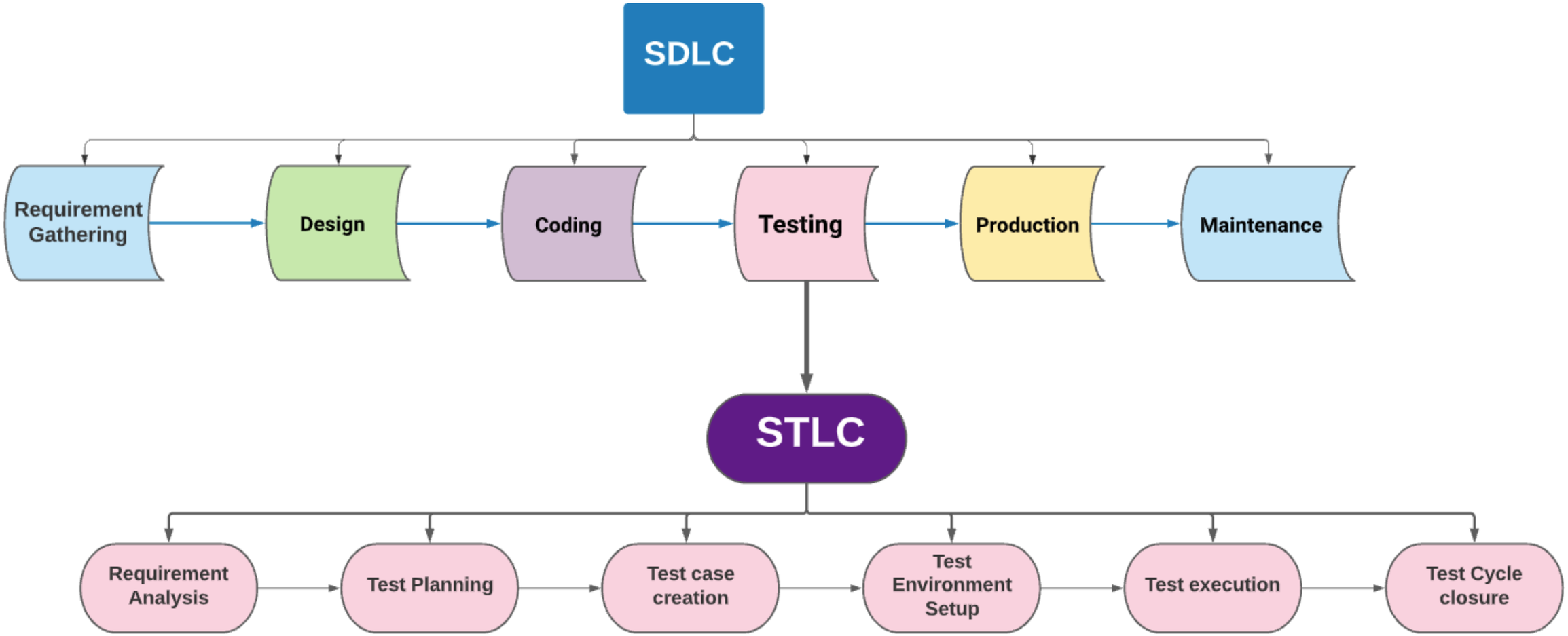
STLC - Software Testing Life Cycle

What is STLC?

- STLC stands for **Software Testing Life Cycle**
- ***Systematic*** *and* ***well-defined*** *steps performed during the testing of a software application*
- STLC helps in a better understanding of the application which results in overall better testing
- STLC also helps to identify bugs from the app and report it


SDLC
Requirement Gathering
Design
Coding
Testing
Production
Maintenance
STLC
Requirement Analysis
Test Planning
Test case creation
Test Environment Setup
Test execution
Test Cycle closure

Step 1: Requirement Analysis

- High level ***requirements for per release is delivered and explained by the Business team - PM, PO, BA***
- During this phase, the QA team may interact with **the Business team** to understand the requirements in detail.
- In some IT company, the Business team just conducts a meeting with the QAs to inform the QAs about what features will be tested, which testing types will be performed with the tools and the release date. QAs learn overall testing methods in this meeting.

- In some IT company, the Business team provides a Test Strategy document to the QA team.
- The project manager(PM) writes a **test strategy** document.
- A **test strategy document** is written to explain **the feature**/s that will be developed and tested, the testing levels and the **types of testing** that need to be performed, and the t**esting tools** to use.

Generally, a test strategy document contains below sections:

Sample Test Strategy link: click here

NOTE:

1. Not all the IT companies prepare test strategy doc.
2. Test Strategy part maybe included in the test plan doc.
3. Some IT companies do not have both test strategy or test plan doc.
4. Some IT companies, test strategy may be written by the QA team based on a template that Business team provided.
5. At Cydeo, testing strategy is combined in the test plan document.

Step 2: Test Planning

- In ***Agile***, planning testing is handled through a meeting **with or without** a test plan document
- Test Plan document generally **prepared by the QA lead** or Senior QA
- Many scrum teams plan testing for each release.
- It is common that there is a master test plan template in the companies that can be used for each project.

Click here to see a sample test plan document

- In test planning, testers determine:
    - Who are the **testers**?
    - What is the role & **responsibility** of each tester?
    - What is the test **strategy**?
        - Which **features** will be tested by the next release?
        - Which **testing types** will be performed?
        - In which **Environment** that testing are performed?
        - What are the **testing tools** will be used? any tool needs to be updated or rollback?
    - What the is **due date**? /**Schedule**?
    - What are the software and hardware **resources**? any app - jira, AWS, VM, etc
    - What is the **input** and **output**? or **entry** Criteria & **Exit criteria**?

Components of A Test Plan	
Items in test plan template	**What do they contain?**
Introduction	Purpose of the test plan document Project brief introduction
Test strategy	Testing framework - BDD cucumber framework Scope of testing - Testing types list & overall info Environments
Schedule	Sprint cycles -starting & ending date
Roles & Responsibilities	Team members are listed How to track bugs? - e.g: jira Who is going to be in charge?
Resources	Software lists - e.g: VM, jira, github Hardware list - e.g: HP
Risks & Mitigation	Potential / possible risks are listed What to do in case of problems?
Approvals	The Names and Titles of all persons who must approve this plan

Test plan VS Test Strategy

Test Plan	Test Strategy
it is project level term	it is organizational level term
carried by QA lead or Testers	carried by Project Manager
It describes how to perform testing project in details manner.	In one company there might be more than one projects, and strategy is done by company level for all projects.
Test Plan includes: what to test?Which feature? how to test? when to test ? Schedule? who will do what test? Role & Responsibilities How will the functional testing occur? Manual or automation? How many cycles of testing will there be? How will the performance testing be coordinated?	Test Strategy includes: What is high level testing objectives? What types of testing that are in scope? What are the risk? What is finish line/. release due date?

Step 3 : Test case creation

- Test case is a document

- QA creates test case doc for per requirement/user story

- There is at least one Test Case for one user story
- Test case document is used for both manual & automation testing
- **A Test Case document has:**
  - Test case ID
  - Test case description /summary
  - Test data
  - Test Environment
  - Test steps
  - Actual & expected results
  - Test pass / fail result
- Creating automated **test script** / code is also done in this step

Test Case ID	ID is a number, unique for each test case.
Description / Scenario	What to test? A phrase/sentence that describes the functionality or feature that the test is verifying;
Related story	Links to user stories
Test Data	Any input that need to test a function. QAs need valid data and/or Invalid data. Test Data is given by developers, or generated by testers, depends on the req.
Environments	The QA environment link that QA will use to test the function
Test Steps	Detailed descriptions of the sequential actions that must be taken to complete the test.
Expected Result	An outline of how the system should respond to each test step; QAs get it from the AC
Actual Result	The result on the QA environment, how the app is actually respond. This section is written after the test execution, not while writing the test case.
Test status	Test status is pass or fail. if expected result == actual result -> pass. if expected result != actual result -> fail. This will be identified after the test execution, not while writing test case doc.

Some other sections for a test case:

Pre-Condition	Any conditions that are necessary for the tester or QA engineer to perform the test.
Comments	Any Scrum members leave comments / feedback.
Bug Report number/link	If a test case fails, then QA creates a bug report; to track the reports easy, the bug reort number or link can be linked to the test case.

A QA can write a good test case with the blow mindset:

- It should be simple and easy to understand

- Avoid repetition

- Additional test setup information can be provided like browser types, operating system, etc.

- It should be written in a way that QA tests only one thing at a time, with no overlaps.

- Ensure that all scenarios, positive and negative, are covered, giving us 100% coverage.

- Test cases should be created with the end-user in mind.

Test Case sample in Excel: Click here

Test Case ID	Scenario/ Descriotion	Environment	Steps	Test Data	Expected Result	Actual Result	Test Status pass/fail
TC01	Verify users login with **valid** data	https://qa1.my.cydeo.com/	1. Go to the login page. 2.Enter valid User name 3.Enter valid password 4. Click on the Login button	username: user1, user34, user56 password: abc123	User should be able to log in		
TC02	Verify users login with **invalid** data	https://qa1.my.cydeo.com/	1. Go to the login page. 2.Enter invalid User name 3.Enter invalid password 4. Click on the Login button	username: abc, uer, usr, 123 password: 432, 134, abc	User should see warning message "invalid username or password"		

User Story:
As a user, I should be able to login.

AC1: Users should be **able to login with valid credentials**.
AC2: Users **should see warning message** "invalid username or password" with invalid credentials

CYDEO

Step 4 : Test Environment Setup

- Testers should make sure all the new codes are deployed from the Dev env to the QA/test environment
- Testers manually and automatically perform testings in test/qa environment.

- **Dev environment ->** Unit & Integration for the developers
- **QA Environment -->** Functional & Non-Functional -> for the QA and performance testers
- **staging Environment ->** UAT -> the client, UAT team
- **Production Environment** -> End users

Step 5 : Test Execution

- Once the user story is clear, test cases are written, test data & environments are ready, testers start manual testing and automation testing by following the TC.
- Test execution means both manual testing & automation testings are performed
- **Expected result vs actual result**
- If expected and actual results are NOT match, testers follow the **Bug Life Cycle**
  - Testers write bug reports
  - Bugs should be fixed by the developers
  - Testers track the bugs, once a bug is fixed, a tester has to re-test to confirm the bug is removed
- Update the test cases with pass / fail results

What is bug life cycle?

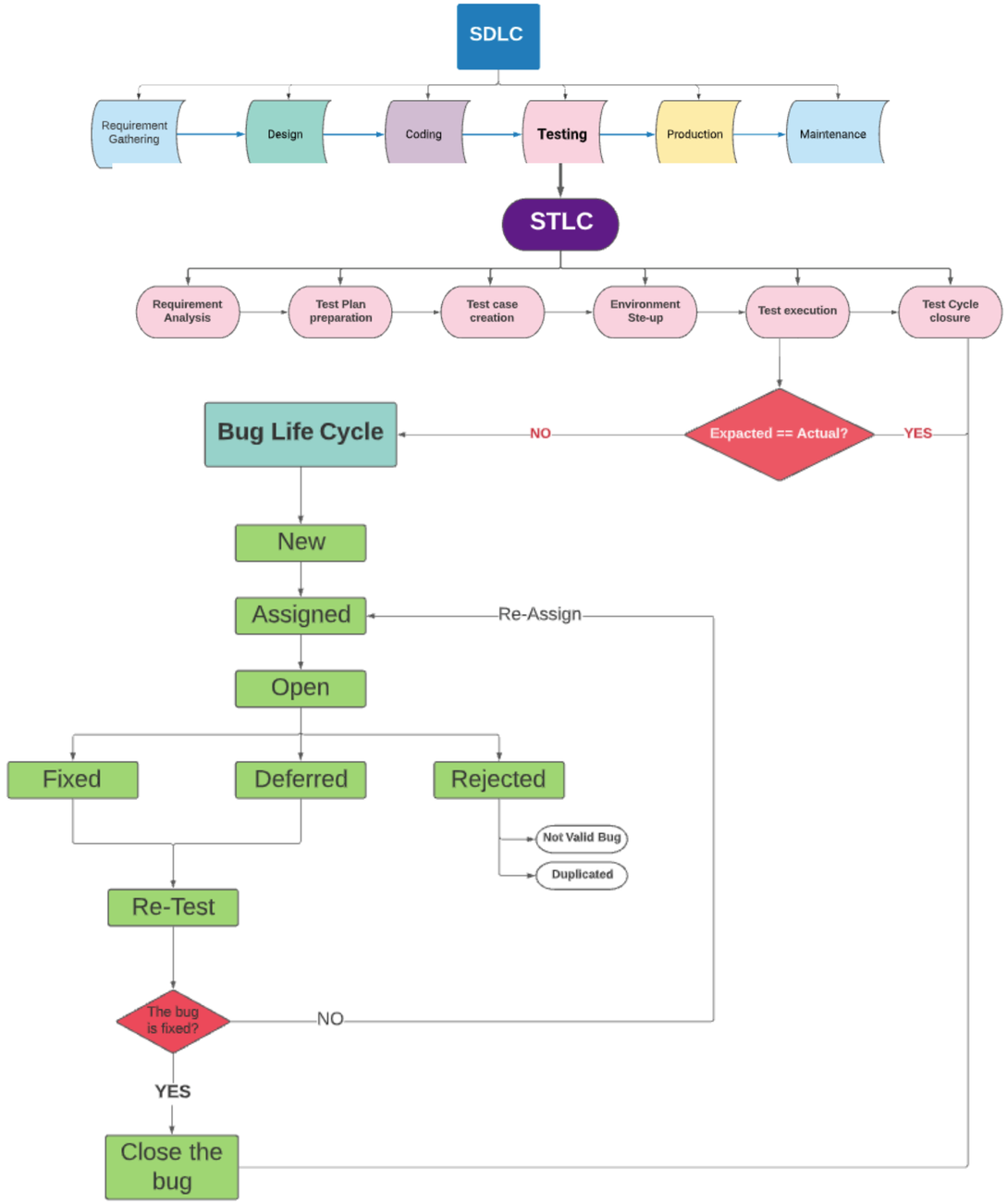
- It is **standard steps** to **handle bug** or defect of a software

- The Bug Life Cycle **includes** specific **set of status** of a bug go through during its life cycle.

- Team members can easily coordinate and communicate current status of defect by the bug status

Bug status are vary from team to team, but generally bugs status are:

1. New
2. Assigned
3. Open
4. Fixed
5. Rejected
6. Deferred
7. Re-Test
8. Closed


SDLC
Requirement Gathering
Design
Coding
Testing
Production
Maintenance
STLC
Requirement Analysis
Test Plan preparation
Test case creation
Environment Ste-up
Test execution
Test Cycle closure
Expacted == Actual?
NO
YES
Bug Life Cycle
New
Assigned
Re-Assign
Open
Fixed
Deferred
Rejected
Not Valid Bug
Duplicated
Re-Test
The bug is fixed?
NO
YES
Close the bug

Bug Status:

1. New: When a tester **finds a new bug**, s/he creates a bug report so the developers can reproduce and fix the bug. In this state, the status of bug is ‘ NEW’.

2. Assigned: Once the new bug has been filed, respective lead (Project Manager/Business Analyze/ PO/ Test Lead) will approve it and **assign the bug to the corresponding developer**. After the bug has been assigned to someone, its status changes to ‘ASSIGNED’ .

3. Open: The development team starts analyzing and works on the defect fix

5. Fixed: When a bug is resolved or fixed by the developer, its status changes to ‘FIXED’ and it is assigned back to the testing team.

6. Deferred: Sometimes a NEW or ASSIGNED bug is given ‘DEFERRED’ status based on urgency and criticality of the bug. A deferred bug’s fix is deferred for some time (for the upcoming sprint).

Bug Status:

7. Rejected: If the assignee (Project Lead/ Project Manager/ Test Lead) or developer finds the bug to be invalid, it is given ‘REJECTED’ status.

9. Re-Test: Tester does the retesting of the code at this stage to check whether the defect is fixed by the developer or not and changes the status to "Re-test."

10. Closed: After the bug is verified, if the bug is no longer exits then the status of the bug will be assigned as “Closed”

Bug Report

Bug ID: #12

Bug status: New / assigned / open / fixed / rejected / deferred / re-test / closed

Reported by: Jimmy (jimmy@compnayName.com)

Date: mm/dd/yy

Assignee: a developer will be assigned

Severity level: high/medium/low

Priority Level: high/medium/low

Bug Description: Students cannot click “login” button on the login page

Bug reproduce steps:

1. Students go to login page
2. Enter valid username & password
3. Click “login” button
4. Verify that students successfully login and launched to the homepage

Expected result: Users should be able to click the login button

Actual result: Users are not able to click the login button

Test data: **student usernames:** student1, student55, student21

Password: 123abc

Environment: URL of the environments

Browser: Chrome, Firefox

Attachments: screenshots and/or video by testers

G03-130 After clicking the "Save changes" button, a spinning wheel appears, and loading never completes.

Details

Type:	Bug
Priority:	↑ High
Labels:	None
Sprint:	Sprint 2 - library
Severity:	High

People

Assignee:

Reporter:

QA Env: https://library2.cybertekschool.com/login.html
OS: macOS Big Sur Version 11.5.2 (20G95)
Browser: Google Chrome 93.0.4577.63

Login credentials:
Login: librarian51@library
Password: Sdet2022*

Step to reproduce:
1. Login as Librarian.
2. Click on module "Books" on top of left navigation menu.
3. Click on "Add Book" button on top of right under navigation menu.
4. Enter "Book Name", "ISBN, "Year", "Author" and "Description".
5. Click on "Save changes" button.

Expected result:
Loading completes, and new book added to the library database.

Actual result:
Page not responding. A spinning wheel appears, and loading never completes.

Step 6 : Test Closure

- Elevates each user story and its Test cases to ensure whether they are executed successfully or not, and manages reports for the future references.

- There are many reports are auto generated by project management app like Jira.
- **RTM** -> Requirement Traceability Matrix document can be prepared.

- **Test cases result reports includes:**
    - How many total number of test cases are created for a user story ?
    - How many test cases are executed? How many manually ? automatically?
    - Total number of bugs ? how many of them are fixed ? how many of them aren't fixed?
    - Total number of test cases per tester executed? sometimes shows the spent time as well

- In agile-Scrum, reviewing the test case reports can be done between the team lead and the business team, or among the testers in the retro to learn from the lesson.

Requirement Tractability Matrix (RTM)

RTM - Requirement Traceability Matrix (RTM) is a document that shows if each requirement has a respective Test case/cases to make sure if the requirement is covered for testing.

Advantages of Requirement Traceability Matrix (RTM):

1. Gives Overview of ALL the requirements
2. Shows how requirements are linked to test cases 3. Makes sure 100% coverage of requirements
4. Easy to prepare
5. No special tool is required, excel, jira etc.

A sample of RTM doc: click here

Online Flight Booking Application's RTM							
Business Requirements Document BRD		Functional Specification Document FSD			Test Case Document		
Business Requirement ID#	Business Requirement Discription	Functional Requirement ID#	Functional Requirement Discription	Priority	Test Case ID#	Execution Status	Defect#
BR_1	Reservation Module	FR_1	One Way Ticket booking	High	TC#001 TC#002	pass	
		FR_2	Round Way Ticket	High	TC#003 TC#004	pass	
		FR_3	Multicity Ticket booking	Medium	TC#005 TC#006	Fail	Defect# 4523
BR_2	Payment Module	FR_4	By Credit Card	High	TC#007 TC#008 TC#009	pass	
		FR_5	By Debit Card	High	TC#010	pass	
		FR_6	By Reward Points	Medium	TC#011 TC#012 TC#013	pass	

RTM in Jira-Xray == Text Execution Report

- A QA team who use the Jira XRAY plugin can get the RTM automatically.
- The Xray generates XRAY reports to track the test cases, its execution status, and bug report of each user story.
- Xray has Test execution report which gives a detailed report of test cases and requirements.
- Test Execution report, or the RTM in Jira-Xray shows how many test execution has created in sprint, how many test total tests are there in 1 test execution, each TC pass or fail status, environment, manual - automation, spent dates, etc are generated.

Test Execution report in Jira-Xray

						Tests By Status						Tests By Test Type							Linked Defects	
TE KEY	SUMMARY	VERSION	PLANNED START DATE	PLANNED END DATE	TEST ENVIRONMENTS	TOTAL TESTS	PASS	TODO	EXECUTING	FAIL	ABORTED	MANUAL	CUCUMBER	GENERIC	OTHER	PROGRESS	SUCCESS RATE	ELAPSED TIME	OPEN	CLOSED
AZUL-343	As a user, I should be able to add news under Company-EU-5 _Test Execution		-	-	HTTPS://QA.AZULCRM.COM/	7	6	0	0	1	0	0	7	0	0	100%	85.7%	-	0	0
AZUL-333	Login Functionality test execution		-	-		1	0	0	0	1	0	0	1	0	0	100%	0%	-	0	0
AZUL-313	AZULCRM-SMOKE_TEST_EXECUTION		-	-		10	10	0	0	0	0	10	0	0	0	100%	100%	-	0	0
AZUL-309	As a user, I should be able to use functions under Employee menu-US21		-	-		7	6	0	0	1	0	7	0	0	0	100%	85.7%	-	0	0

CYDEO