Template-Test Strategy Standards

Contents

Flight Profile Optimization (FPO)
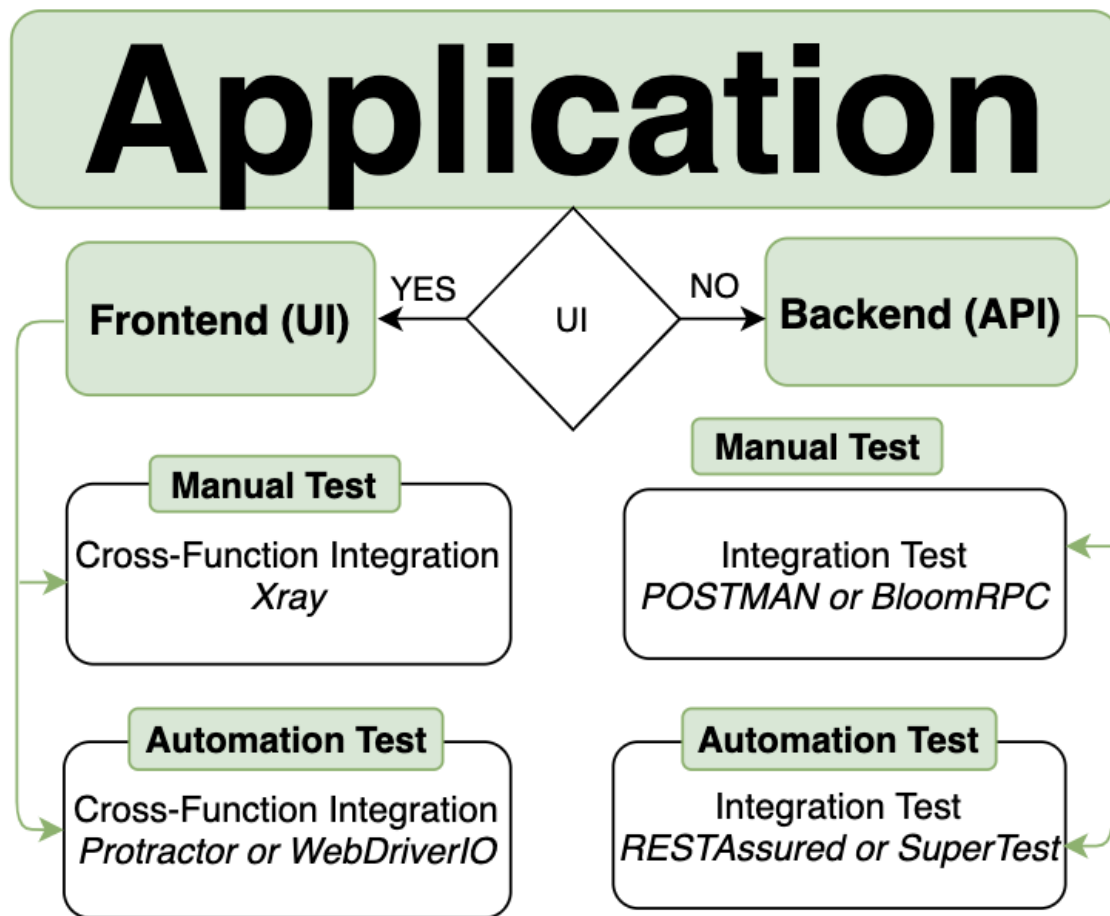
Customer Vision & Problem Statement

For pilots and airlines who need to improve the comfort and convenience of their passengers' travel experience, Collins FPO is a sophisticated, NASA-tested route optimization tool that offers efficient traffic- and weather-aware in-flight reroute solutions for a faster, more fuel-efficient flight.

Domestic airlines in the U.S. spend an estimated combined $2 to $5 billion on jet fuel every month. JetBlue and other airlines are looking for an Electronic Flight Bag (EFB) solution that helps them optimize fuel and time on their flights to reduce operational costs and improve the passenger experience.

Listed below all the important features to be tested!

JIRA Epic & User Story (Features) Link

1. Route suggestions
2. Basic map display
3. Map navigation
4. Startup errors
5. Set optimization objective
6. Connection errors
7. Map layers
8. Data feeds display
9. Flight status
10. Set altitude limit
11. Set maximum deviation limit
12. Set optimization limit
13. Startup checklist
14. Record ATC & dispatch response

List of Testing Types of a single application

Application

Frontend (UI) ← YES — UI — NO → Backend (API)

Frontend (UI)

Manual Test
Cross-Function Integration
Xray

Automation Test
Cross-Function Integration
Protractor or WebDriverIO

Backend (API)

Manual Test
Integration Test
POSTMAN or BloomRPC

Automation Test
Integration Test
RESTAssured or SuperTest

In-General Dev vs QAE ratio rules

- 2 to 4 Developer (Front-end & Backend) = 1 Manual QAE + 1 Automation QAE
- 5 to 6 Developer (Front-end & Backend) = 2 Manual QAE + 1 Automation QAE
- 7 to 8 Developer (Front-end & Backend) = 3 Manual QAE + 2 Automation QAE

Why No Test at all?

1. Staging Environment (Mimic of the production): If there is no staging (QA) environment available to test the application and testing on the development environment which is functionality frequently changing by the developers (not able to perform regression test at all) in that case testing may be a waste of time because functionality needs to be tested again on the Staging Environment.

Features not to be tested

1. Hardware

What to automate:

It consists of the area of your Application Under Test which you are planning to automate, like:

1. Is the application stable enough to automate in a stable environment?
2. Common functionality throughout the application
3. When a large amount of data is used as Test Inputs & Condition

4. Business-critical functions (Complex Test Cases)
5. Testing that needs to be repeated often (Re-testing/Regression Testing)

What Not to Automate

It is not always advantageous to automate test cases. There are times when manual testing may be more appropriate. For instance, if the application's user interface will change considerably in the near future, then any automation might need to be rewritten anyway. Also, sometimes there simply is not enough time to build test automation. For the short term, manual testing may be more effective. If an application has a very tight deadline, there is currently no test automation available, and it's imperative that the testing gets done within that time frame, then manual testing is the best solution.

1. Unstable Functionality
2. Design
3. Low return on investment
4. Unstable environments - unable to handle high volumes of tests running in parallel; application infrastructure not repeatable and disposable, or environment not sized correctly.
5. Too many networks hop - causing timeouts or unknown issues.
6. The native mobile application isn't designed for testing. Developers don't understand the locator strategy (accessibility id).
7. Poorly written test scripts - no coding standards (unit test) or design patterns.
8. Data collision - no unique data or settings.
9. UI testing everything.
10. Lack of communication on software changes.
11. No quality ownership from developers.
12. resource constraint problem

How to trace? User story to be automated in JIRA:

- **JIRA Label* - Required_QA:** During the Refinement/Sprint Planning meeting, the Product Owner will decide the story needs to be testing or not? based on that decision product owner/Scrum master/TPM will add the Label on the JIRA user story.
- **JIRA Label* - Required_Automation:** During the Sprint Planning meeting QA Automation Engineer should decide the user story will be automated or not? based on that decision product owner/Scrum master will add the Label to the JIRA user story.
- **JIRA Label* - Automation_Done:** After Automating the Manual test cases of User Story, QA Automation Engineer should add the Label "AutomationDone" and if it is partially developed it should remain with the comment what's the reason behind for Partially Automated.
- **JIRA Label* - ManualRegression (If you don't have automation)**
  - If the "User Story" doesn't meet the Automation Criteria, SQA Engineer should add the Label "ManualRegression".

Scope of Testing

Xray - Test Management Tool Including RTM Link (Test Case Coverage)

Types of Testing	Yes /No	Testing approach and Tools	Comments
Integrated Functional Test (UI)	Yes	Manual (Xray) & Automation Framework-Protractor	For Mobile Automation: Frontend Automation (CI/CD) Only possible! 1. If we have SauceLabs real device license 2. If we are allowed to deploy the app on Saas Environment
Integration Test (API)		Manual (POSTMAN) + Automation (RestAssured OR SuperTest)	
Regression	Yes	XRay, RestAssured For API, CI/CD (Jenkins)	
Performance Test	Yes	Load & Stress	Based on the Requirement - SLA
Compatibility	Yes	Chrome, Firefox, IE, Edge	
Localization & Globalization	No	[Translation, Date time format, Number format]	
Security	Yes	Run by Esia Yosupov	

Bug/Defect Tracking

JIRA - Bug tracking tool Link

- JIRA will be used as a defect tracking tool across the project.
- Defect Priority will be initially assigned by the QA team and the Product Manager/Owner can modify.

Success / Exit Criteria

- All the approved functional and Regression test cases/script should be executed and passed
- No Blocker, Critical or Major Severity Bugs should be open

- Business Users should perform UAT and Sign off
- Deferred bugs need to be approved by Business and Development teams
- All test results and corresponding defects are documented and reported to stakeholders.

Phase Sign-Off Responsibility

Role	Location	Name
QA Full-Stack Engineer	DX-Brooklyn	MD Zaman. (40%)
Product Manager	DX-Brooklyn	Anya Palkowski

Tracking Testing Progress

- Manual - Xray Test Management Plugin in JIRA
- Automation - Jenkins (CI/CD) Pipeline through the GitHub Repo

Task distribution – Define roles and responsibilities of each team member

- MD Zaman - (100%) - QA Full-Stack Engineer

Testing Report

- Manual - Xray Test Management Plugin in JIRA
- Automation (API) - Will Generate HTML Report and send it to a team email through the CI/CD build.

What All The Risk During The Test Completion?

- Delay of Test Environment setup (QA/Stage)
- Incomplete environment to support testing (Unstable test environment)
- Delay of CI/CD Process
- Delay of Device Environment IPA/APK (Simulator/Real Devices)
- Delay of the physical device for QA & Dev
- Delay of the lockdown the Functionality
- Dedicated resources unavailable (Not enough resources)
- Delay of Service (Test) Account (if application support single sign-on)

Related Software Design or Other Documents

- FPO- Frontend (UI) & Automation GitHub repo
- FPO - Backend (API) & Automation GitHub repo
- Xray - Test Management Tool Including RTM Link (Test Case Coverage)
- JIRA - Bug tracking tool Link

Test Environment

List all necessary environments

- QA/Staging environment

Exclusions

Browser/Devices	OS/Browser Version	iOS/Browser Version	~~Android Version~~
iPad	13.2	7th Generation	~~N/A~~
Chrome	Latest	N/A	N/A