17th May 2020

Random Date Generator – Test Strategy

Table of Contents

1. Introduction 4

1.1. Project Overview 4
1.2. Purpose 4
1.3. References 4
1.4. Scope 4
1.4.1. In Scope 4
1.4.2. Out of Scope 5
1.5. Assumptions 5

2. Test Strategy 6

2.1. Overall Approach 6
2.1.1. Functional Test Validation 7
2.1.2. Test Automation 9
2.2. Test Schedule 10
2.3. Test Process 10
2.4. Test Bed Setup 12
2.5. Defect Reporting 12
2.6. Control Procedures 12
2.7. Risks 13
2.8. Dependencies 14
2.9. Tools for Testing 14

3. Test Environment Requirements 15

4. Communication 15

4.1. Defect triage meetings 15
4.2. Daily scrum/standup meeting 15
4.3. Test Report 15

5. Entry and Exit criteria 16

5.1. Test Planning 16
5.2. Test Design 16
5.3. Test Execution 16

6. Roles and Responsibilities 17

7. Release mechanism 17

7.1. Acceptance criteria 17
7.2. Deliverables 18
8. Acronyms 18

1. Introduction

1.1. Project Overview

Random Date Generator (RDG) is a web application that allows you to generate random calendar dates, based on different criteria. It allows you to select different date format and even customize the output. Some of the key features are:

- Add to favourites
- Save and share
- Generate random dates (different output format available)
- Can generate up-to 9999 random dates
- Can customize output date format

1.2. Purpose

The Purpose of this document is to define the following from testing perspective.

- Scope and out of scope items
- Test strategy
- Activities required to prepare for and conduct UI, Integration, system, regression and acceptance testing
- Various dependencies, assumptions and risks
- Test environment requirements
- Entry and Exit criteria
- Test deliverables

1.3. References

S. No	Location
User stories	<<reference>>
Test Plan	<<reference>>
Test Cases	<<reference>>
UX/Wire Frames	<<reference>>
FSDs	<<reference>>

1.4. Scope

1.4.1. In Scope

The RDG will be tested on Desktop browsers. Following functional modules are in-scope for RDG. The systems will have a multi-lingual capability for all the screens and interfaces. <<List of languages to be added later>>

- Add to favourite
- Save and Share

- Random date Generator form
  - Predefined output date format
  - Custom output date format
  - Start & End date
  - Number of dates
  - Text area to display generated dates
  - Generate Random Dates

From the testing perspective following items are identified as in-scope.

- Testing the functionalities of RDG as per the acceptance criteria mentioned in User Stories.
- UI Verification on different browsers as per acceptance criteria mentioned in User Stories.
- Build Verification testing
- Automation of identified test cases

Note: Please refer <<Test Environment Requirements>> for the browsers that are in-scope

1.4.2. Out of Scope

- Security and compliance testing
- Testing on third party services if any
- Database testing
- Web service testing of the existing RDG APIs
- Mobile Testing
- Performance testing

1.5. Assumptions

- The test cases will be prepared based on the approved and signed off User stories and wire frames.
- Test coverage is bound to functional/user specification
- Builds will be delivered to QA on schedule for environments planned to setup and should be released with
  - Installation notes if any
  - Release notes
  - Unit testing results
  - Known issues/limitation if any
- Dedicated testing environment will be ready and usable
- Functional testing will take place on dedicated Test environments
- Testing will be carried out in the specified browser versions
- All test cases will be peer reviewed
- The defects reported by the test team will be logged in Defect tracking tool and sufficient information will be provided including steps to reproduce and screen shots.
- No further changes or inclusions will be considered for inclusion in the released version except in cases where (1) change is critical and has approval from stakeholders (2) changes/inclusions will not require significant effort on behalf of the test team and will not adversely affect the test schedule. Any major changes to design will entail additional time to re-plan testing effort

2. Test Strategy

Scrum/Kanban methodology will be adopted where progress will occur in an iterative and incremental manner and be divided in to two-week cycles called sprints. The test approach is to exercise all testing techniques mentioned below on RDG and uncover as many defects as possible.

The Testing phase will involve the below activities in every sprint cycle.

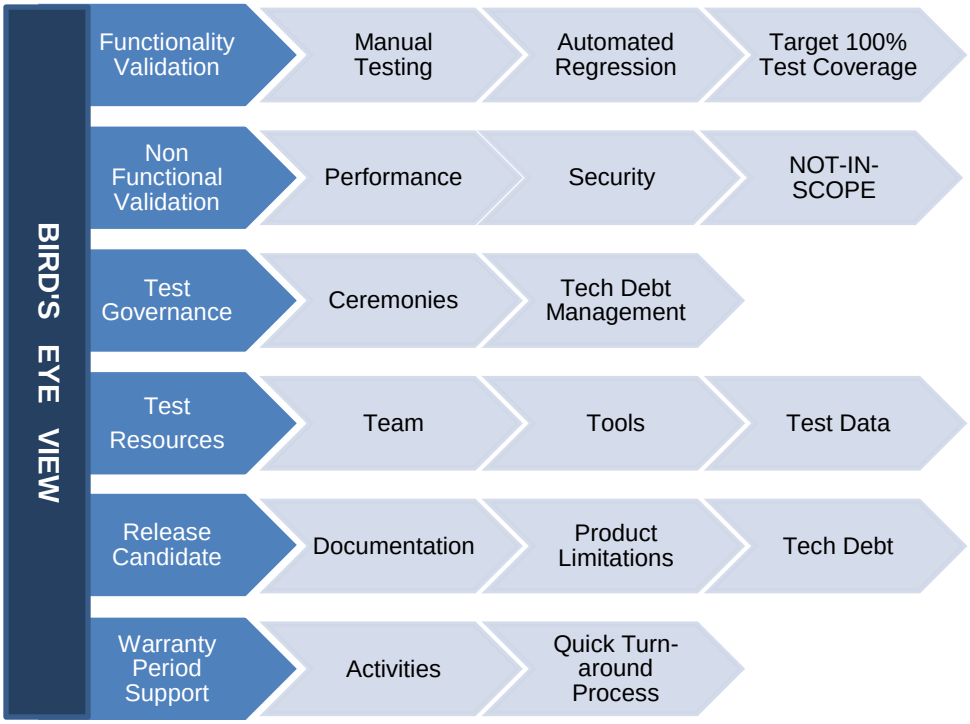
a) Daily Scrum Meeting
b) Sprint Planning
c) Grooming of User Stories
d) Test cases preparation and review
e) Test environment setup
f) Test execution(Manual and Automation)
g) Defect report
h) Test execution report
i) Identification and test script development of automatable test cases

Following test techniques would be undertaken during Test Execution in Web

a) Build Verification Test (BVT)
b) Functional Testing
c) Integration Testing
d) System Testing
e) Regression Testing

2.1. Overall Approach

Below diagram provides a glimpse of the overall Test approach from inception to launch.



2.1.1. Functional Test Validation

- Test procedure design and prioritization- Test cases of RDG will be prioritized into different categories - P1, P2, and P3 of the overall test cases designed.
- Test cases generated will be based on business scenarios, wire frames and user stories. This means that each test case may have more number of verification points.
- For feature/functional testing, BVT and all the test cases pertaining to that feature will be executed.
- For system testing – BVT's, P1's, P2's and P3 test cases of RDG, Exploratory testing and Regression testing will be executed on standard Test environment.
- On bug fixes, retesting and impact based testing will be done and the defect will be closed / reopened.

Testing Methodology

Build verification Testing (BVT) Approach

Test Objective	To verify if the build is testable with critical business flows and high priority functionality is working as expected.
Technique	• Identify all major workflows and scenarios critical to quality as BVT • When a new build is released to test, test team will execute the BVTs and certify if the build deployed can be accepted for further testing
Potential Tools	Manual and Automation testing
Success Criteria	100% pass rate of identified scenarios and workflows.

User Interface Testing Approach

Test Objective	To Verify • Navigation through the application properly reflects business functions and requirements, including field to field, and use of access methods (tab keys, mouse movements, accelerator keys) • UI components and characteristics, such as menus, text areas, tables, size, position, state, and focus conform to standards. • Verify the components on different browser versions
Technique	• Create tests for each field to verify proper navigation and object states. • Verify there is no UI distortion on different browsers
Potential Tools	NA
Success Criteria	Each screen successfully verified to remain consistent within acceptable standard.

Functional Testing Approach

Test Objective	Ensure all functions of the system works in accordance with the agreed requirements/use cases.
Technique	Execute each use case using valid and invalid data, to verify the following: • The expected results occur when valid data is used • The appropriate error / warning messages are displayed when invalid data is used • Each business rule is properly applied
Potential Tools	NA
Success Criteria	• All planned tests have been executed and passed • All identified defects have been addressed and fixed defects are retested

Integration Testing Approach

Test Objective	Ensure that the interaction of all the components produce results that satisfy functional requirements.
Technique	Execute test component with valid and invalid flows, to verify the following: • Data output of one component to another component(s) • Workflow connecting two or more components or application
Potential Tools	NA
Success Criteria	• All planned tests have been executed for all components linked to each other. • All identified defects have been addressed and fixed defects are retested.

System Testing Approach

Test Objective	Ensure that the system satisfies the functional and non-functional requirement as in the approved User Stories and any inputs and outputs generated from or received to the application under test.
Technique	Execute test component with valid and invalid flows, to verify the following: • Execute test components with valid and invalid inputs • End to End business flows
Potential Tools	NA

Success Criteria	• All End to End tests have been executed and passed • All identified defects have been addressed and fixed defects are retested.

Regression Testing Approach

Test Objective	Ensure that previously detected and fixed issues are functioning properly. They do not reappear and new issues are not introduced into the program due to defect fixes.
Technique	Use Test cases developed for: • Build Verification Testing • User Interface Testing • Functional Testing • Integration Testing
Potential Tools	NA
Success Criteria	For each combination of component and workflow software, all transactions are successfully completed without failure.

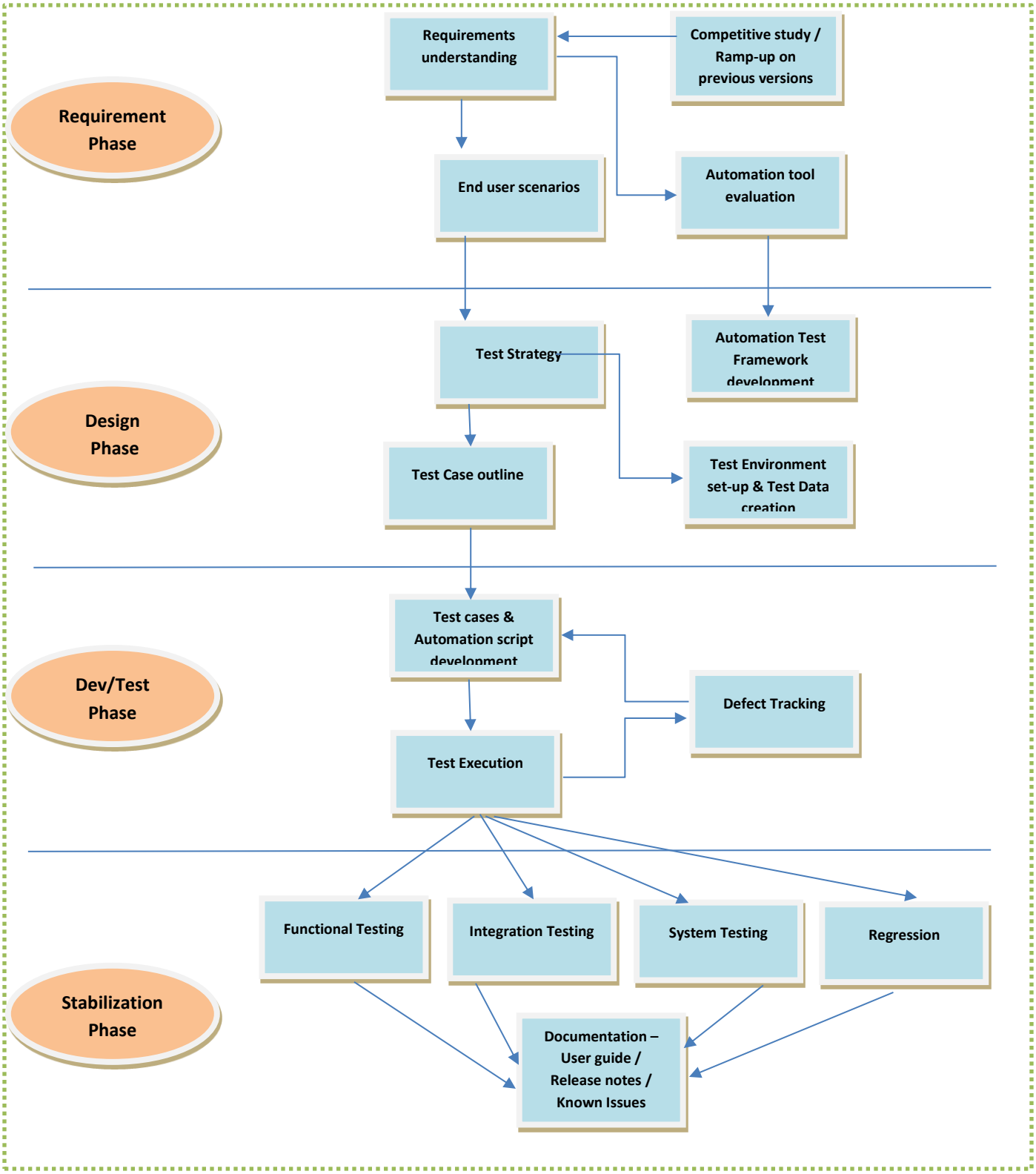
2.1.2. Test Automation

Test Objective :	Automation of functional test cases is aimed at reducing the manual execution
Technique:	• Automation candidates will be identified for each sprint. • Test cases of N-1'th sprint will be taken for automation • Automate 100% of automatable BVT Test cases • P1 and P2 test cases will be considered for automation based on the timelines and once BVT are exhausted • Stabilization all the test cases • Regression pack will be prepared with a group of business priority test cases and will be executed when major bug fixes/releases are introduced. • Automation scripts will be designed to run on desktop browsers
Potential Tools:	• Java • IntelliJ • Selenium • Cucumber • Maven-Cucumber reports

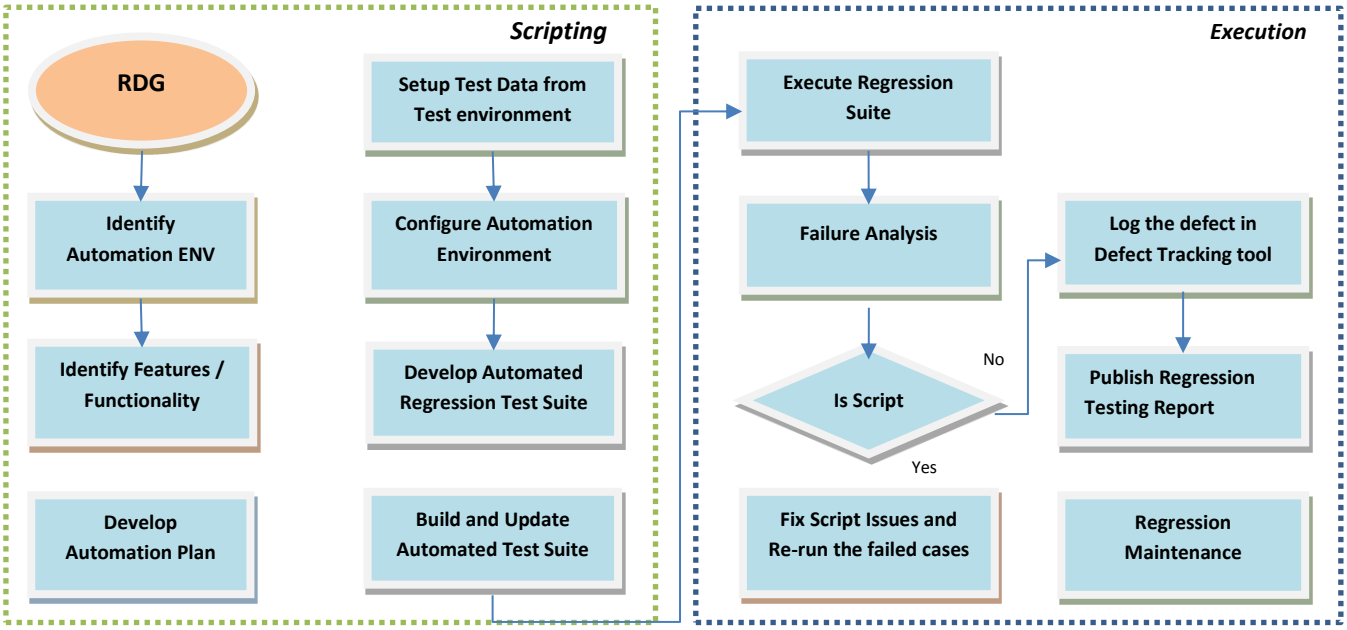
2.2. Test Schedule

Test Schedule will be planned sprint wise and will be maintained in Test Management Tool. Please refer this <<link>> for test schedule.

2.3. Test Process



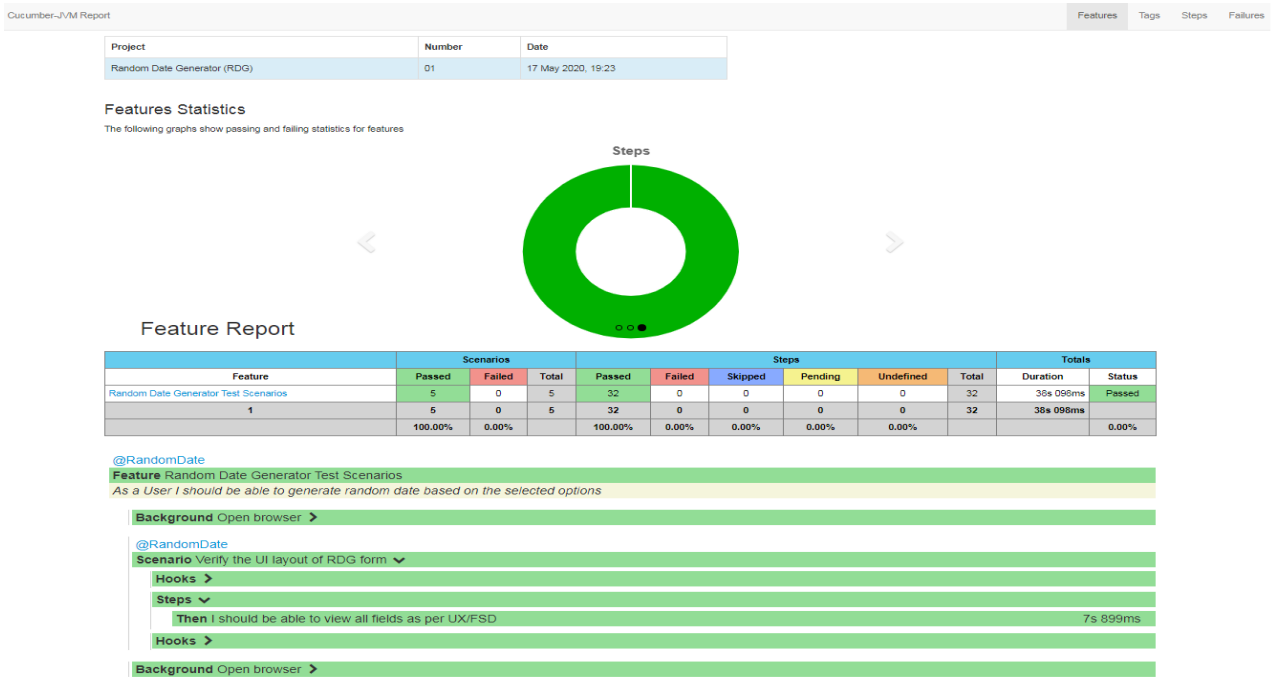
Automation Test validation Approach



Automation Framework

<< To be added later>>

Automation Reports



2.4. Test Bed Setup

<< TBD >>

2.5. Defect Reporting

Severity Level	Definition for the Severities
Critical	The defect affects critical functionality or critical data. It does not have a workaround
High	The defect affects major functionality or major data. It has a workaround but is not obvious and is difficult
Medium	The defect affects minor functionality or non-critical data. It has an easy workaround
Low	The defect does not affect functionality or data. It does not even need a workaround. It does not impact productivity or efficiency. It is merely an inconvenience

Priority Level	Definition for the Priorities
Critical/P0	Any defect that causes further testing of the product to be blocked will be categorized as an urgent defect. e.g.: A Permission not updated on the environment causes testing to be blocked across all features
High/P1	In case an entire functionality that was supposed to be in the build is missing, it will be categorized as a “High” priority defect. This kind of defect does not affect other functionalities to be tested
Medium/P2	• Part of a requirement not working correctly • Any negative flows that are failing • Clarifications, Requests
Low/P3	• Translation Issues • Minor alignment issues • Scripting issues • Any other browser related issues • UI issues

2.6. Control Procedures

Reviews

The project team will perform reviews for each Phase. (I.e. Requirements Review, Test Plan Review, Test Case Review and Final Test Summary Review.

Triage meetings

This meeting will be held to discuss reported defects. The development team will provide status/updates on all defects reported and the test team will provide addition defect information if needed. All members of the project team will participate.

2.7. Risks

SL No	Category	Risk	Impact	Mitigation/Contingency	Owner
1	Requirements	Change in scope	Impact on test time lines	Change Test effort and Schedule accordingly. Reschedule the plan	Test Lead
2	Environment	Availability or Unscheduled environment down time	Impact on test timelines	Alternate environment to be organized by the Scrum master to carry out testing	Scrum Master
3	Build schedule variance	Delay in getting build	Impact on test timelines	Highlight blockages & impact at the earliest to SM	Test Lead
4	Build Quality	Too many defects found during test execution	Delay in test execution completion	Highlight blockages & impact at the earliest to SM. Test team will Identify the impact areas and perform regression testing in these areas	Test Lead
5	Build Quality	Failing of BVT and acceptance scenarios in the stable build	Delay in test execution completion	Build will be rejected. Escalate such issues as high priority for immediate resolution.	Test team
6	Defect fixes	Delay in defect fixes	Impact on timelines	Highlight to SM. Conduct periodic triage meetings to prioritize defects to be fixed	Test team
7	Resources	Test resources unavailability and planned leaves	Impact on test timelines	Back up resources to be planned and work load to be shared based on the criticality	Test Lead
8	Support	Lack of technical support for tools used while testing(e.g.: Jira going down)	Impact on test activities	Ensure technical support for tools used is available whenever required.	Scrum Master

2.8. Dependencies

Sl No	Dependent Actions	Owner
1	Approval and timely feedback of artefacts like Test Plan, Test Cases and test reports	Product Owner
2	All the unit test cases of the RDG are executed completely.	Dev team
3	All the sprint specific user stories / wire frames should be available in the repository before the sprint start date.	Scrum Master
4	All the defects found in Feature Testing and Functional testing will be logged on to Jira. All the resolved bugs in Jira should have the details of the changes made with the changed files name and an email will be sent with respect to bug fix. Bugs resolved without change list details will not be accepted. Test team will verify the bug and change the status based on the result of the tests.	Dev Team
5	Provide impact areas as part of each sprint development to enable QA focus on them in addition to the new functionalities implemented in the current sprint	Dev Team
6	All items should be clarified before the work item begins	Product Owner
7	Regular builds of the application should be available for testing	Dev Team
8	Availability of test environments before test execution	Scrum Master

2.9. Tools for Testing

Sl No	Tool	Tool Type
1	TBD	User stories/ Test Cases / Defect tracking
2	IntelliJ, Selenium, Java, Cucumber	Web Automation
3	TBD	Project Management tool

3. Test Environment Requirements

Channel	Primary	Secondary
Browsers	Chrome	Firefox, IE, Safari
OS	Windows, Mac	-

Testing will be conducted on <<Test environment>> in a sprint, there might be 2-3 builds based on the no. of user stories and defect fixes.

4. Communication

4.1. Defect triage meetings

Defect triage meetings will be held in the following cases

- To be held on daily basis to resolve the disagreement between Product Owner, Dev. & QA
- To be held in case of business priority needs to be assigned
- To be held in case of ambiguity in the application expected behavior
- Suggestions can be discussed in the triage meeting

Audience of Bug triage meeting:

Product Owner, Scrum Master, Dev lead and Test Lead. Other team members will join on need basis.

4.2. Daily scrum/standup meeting

Daily stand up meeting would be conducted to discuss the previous day's progress, plan for the current day and any impediments. All the team members will be part of this meeting.

4.3. Test Report

Test team would send regular reports. These reports would be focusing on general updates

Daily Status Report

Daily status report will be sent by scrum master inclusive of testing tasks at the end of working day highlighting the key activities for the day.

Release Report

Release report would be of combination of all sprints, release related matrixes, Projects/ Functions signed off as part of the iterations and any known defects which are agreed with Product Owner/Business.

5. Entry and Exit criteria

5.1. Test Planning

Scope: Understanding the requirements and preparing test strategy and Plan.

Entry Criteria:

- Final set of requirements/use cases are finalized
- Testing Scope and approach of the project is finalized

Suspension Criteria*:*

- Requirements not finalized
- Change in scope and schedule

Resumption Criteria:

- Unblocked on issues mentioned in Suspension Criteria

Exit Criteria:

- Signed off test strategy and Plan

5.2. Test Design

Scope: Preparation of test scenarios and test cases based on the finalized requirements

Entry Criteria:

- Signed off test strategy and Plan
- All process flows and business requirements are signed off
- All clarifications answered by Product Owner or Business Analyst

Suspension Criteria*:*

- Requirements not finalized
- Awaiting clarifications answered by Product Owner or Business Analyst.

Resumption Criteria:

- Unblocked on issues mentioned in Suspension Criteria

Exit Criteria:

- Test cases reviewed and signed off by Business Analyst and Product Owner

5.3. Test Execution

Scope: Execution of test cases on different environments.

Entry Criteria:

- Test cases signed off by Business Analyst and Product Owner
- Unit testing of the code complete
- Build releases and deployed

- Release Notes sent to QA team
- All Build Verification Tests pass
- All the P1/P2 defects documented during Development phase must be fixed and closed

Suspension Criteria*:*

- High priority test cases failed
- Build crashes & there is no workaround
- Environment unavailable

Resumption Criteria:

- Fixes for high priority defects
- Environment backup

Exit Criteria:

- All test cases have passed
- No open defects. Any deviations from this to be approved by Product Owner /Business
- QA Sign-off for each stage of testing

6. Roles and Responsibilities

Role	No. of resources	Responsibilities
Test/Automation Lead	1	• Primary point of contact • Test planning & strategizing • Test Estimation • Framework design • Risk management • Epic and user stories review • Test review and execution • Scripts/Code reviews, integration • Status & Test Reports
Test Engineer	TBD	• Epic and user stories review • Test Case review • Test Execution • Automate test cases

7. Release mechanism

7.1. Acceptance criteria

The purpose of acceptance test is to confirm that the acceptance criteria of all the user stories are implemented properly and all the features are working as expected. All the below mentioned criteria's should meet for a successful UAT.

- All system tests cases executed.
- No P0 or P1 defects outstanding
- All the functionalities are working as per the acceptance Criteria
- All Acceptance criteria's executed and signed off by product owner

7.2. Deliverables

The following documentation will be available at the end of the test phase:

- Test plan
- Manual Test Scenarios/cases
- Automation test scripts
- Sprint Test report
- Sprint Defect report
- Release Report

8. Acronyms

- UAT – User Acceptance Testing
- TBD – To be decided
- BVT – Build Verification Testing
- UI – User Interface
- RDG – Random Date Generator