Wrisk Test Plan

[1. Document Revision 2](#_Toc482800201)

[2. Introduction 2](#_Toc482800202)

[2.1 Purpose of this document 2](#_Toc482800203)

[2.2 Objectives of Testing 2](#_Toc482800204)

[2.3 Sprint Definition 2](#_Toc482800205)

[2.4 References 2](#_Toc482800206)

[3. Dependencies 4](#_Toc482800207)

[3.1 Test Environment 4](#_Toc482800208)

[3.2 Test Data 4](#_Toc482800209)

[3.3 Test Tools 4](#_Toc482800210)

[3.4 External Services 4](#_Toc482800211)

[3.5 Internal Services 5](#_Toc482800212)

[4. Test Strategy 6](#_Toc482800213)

[4.1 Unit Testing 6](#_Toc482800214)

[4.2 Acceptance Testing 6](#_Toc482800215)

[4.3 System Testing 6](#_Toc482800216)

[4.4 System Integration Testing 6](#_Toc482800217)

[4.5 Regression Testing 6](#_Toc482800218)

[4.6 Exploratory Testing 6](#_Toc482800219)

[4.7 Non-Functional Testing 7](#_Toc482800220)

[4.7.1 Performance Testing 7](#_Toc482800221)

[4.7.2 Load/Stress Testing 7](#_Toc482800222)

[4.7.3 Security Testing 7](#_Toc482800223)

[4.8 Automation Strategy 7](#_Toc482800224)

[4.9 Bug Fixing 7](#_Toc482800225)

[4.10 Testing Principles 7](#_Toc482800226)

[4.11 Scope 8](#_Toc482800227)

# Document History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version No.** | **Author.** | **Revision Details.** | **Revision Date.** | **Distribution.** |
| 0.1 | Tuely Saseendran | Initial Draft of Wrisk Test plan for HH | 12 March 2018 | None |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Introduction

This document describes the overall test strategy for the **Wrisk project**. This plan takes into consideration the agile/scrum methodology that the Platform teams follow throughout this project.

## Purpose of this document

This document outlines the various tests – unit, system, acceptance, integration, regression and non-functional testing.

Among the various modules that are part of the project, this test plan focusses on the **Wrisk Motor Journey**. Approach for executing various phases of testing for the different aspects of Wrisk Motor Journey – such as Vehicle details look up, motor journey in AV, Vehicle registration look up, Branding etc – are detailed here.

## References

Any externally referenced documentation associated with the testing of this product are listed below.



# Dependencies

A list of dependencies associated with testing of the product are outlined in this section.

## Test Environment

The platform team will do testing in development environment. Development team will be responsible for deploying to test and UAT environments.

## Test Data

TBD

## Test Tools

Tools required for executing the various test phases are listed below.

* Visual Studio – The team will use Visual Studio 2017 Enterprise Edition to create and debug test automation for the product
* RoboMongo – Required to interrogate the Mongo DB for transactional data

## External Services

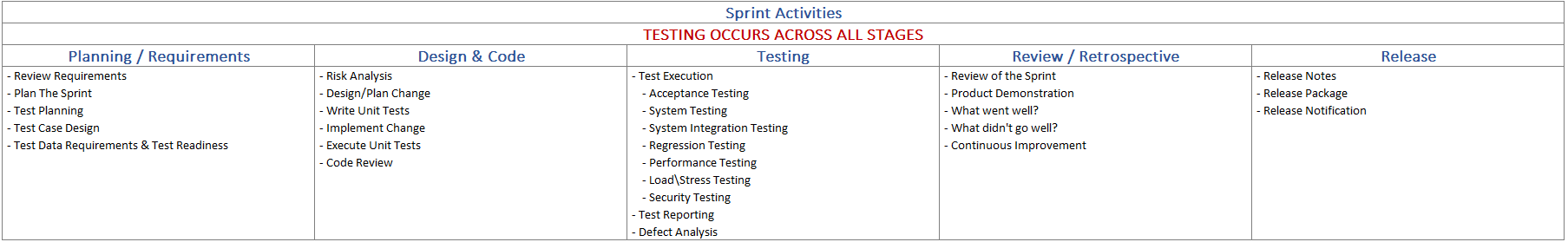
External services that will be used during various test phases are listed below.

* Vehicle Registration lookup - <https://api.dvlasearch.co.uk>
* Stripe Payment Service
* PremiumCredit (Finance)
* Thatcham
* Sanction Check
* ID Check
* Premium Credit
* MID
* Broker Direct

# Test Strategy

The product development/test strategy is agile. A view of the agile and testing methodology is outlined below.





## Unit Testing

Unit Testing shall be completed by the developers implementing code changes. Unit tests should look to exercise the smallest testable parts of the code to scrutinize it for proper operation. Unit testing should always be automated and run at build time.

Scope of unit testing for this project will include the below (subject to further review/design discussions):

* Validate all the mandatory fields are present in MV Motor Journey
* Validate the content of the mandatory fields to ensure they comply with the format and business rules
* Check for presence of any optional fields. If present, validate the content to ensure they comply with the format and business rules
* Verify the stripe response

## Acceptance Testing

Acceptance Testing is the verification and validation of the implementation of a user story or bug based upon the acceptance criteria. Acceptance criteria is written in Gherkin language to avoid any ambiguities. From this acceptance tests are derived to ensure the business requirements are met.

Acceptance criteria for Wrisk will include the below (subject to further review/design discussions):

Feature: Wrisk Motor Journey

* User successfully able to add the vehicle details
* User successfully able to add the vehicle usage details
* User successfully able to add personal details
* User successfully able to view details from Vehicle details look up
* User successfully able to view details from Vehicle registration look up
* System able to calculate and display relevant quotes
* Validate response message from Wrisk - PremiumCredit (Finance) API
* User should get an Email confirmation

Feature: Policy

* System should allow user for Cancellation of polices
* System should allow user for MTA(Mid Term Adjustment) of policies
* System should allow user for Renewal of policies
* System should allow user to do 7 day Policy while entering valid voucher code
* System should allow user to do Policy Add ons (Non insurer options -Legal expenses )

Feature: Motor Rating engine

## System Testing

System Testing is the process of verifying that the integrated product correctly operates and functions with all other components and services within the system.

System testing for wrisk should cover the following message types and data flows (subject to further review/design discussions):

* A successful Motor Journey for wrisk via Agent/client
* An agent/Client should be able to create Free/ Annual/ Monthly policy
* An agent/Client should be able to successfully cancel a policy
* An agent/Client should be able to do Midterm Adjustments(MTA)
* An agent/Client should be able to do successful payments via stripe
* An agent/Client should be able to add Non insurer options(add legal expenses) in policy
* An agent/Client should be able to do cancel Non insurer options from policy
* An agent/Client should be able to do MTA for addons

## System Integration Testing

System Integration Testing is the process of verifying component integrations of a product to verify a compliance with the specified system integration requirements.

System Integration Testing for wrisk should cover the following data flows

* Stripe Payment Integration
* ID Check
* MID
* Sanction Check
* Zendesk Integration
* Tableau Integration
* Mid Integration

## Regression Testing

Regression Testing is the process of verifying existing functionality and requirements within the product continue to function as desired after changes have been made. Regression testing is generally the last phase of the testing process to verify a product release readiness. Where possible regression testing should be automated.

Wrisk project should include the following regression testing scenarios (subject to further review/design discussions):

* A successful Motor Journey for wrisk via Agent/client
* An agent/Client should be able to create Free/ Annual/ Monthly policy
* An agent/Client should be able to successfully cancel a policy
* An agent/Client should be able to do Midterm Adjustments(MTA)
* An agent/Client should be able to add Non insurer options(add legal expenses) in policy
* An agent/Client should be able to do cancel Non insurer options from policy
* An agent/Client should be able to do MTA for addons

## User Acceptance Testing (UAT)

User Acceptance Testing is the responsibility of the Implementations team, who will complete this as part of the New Product Inspection process (NPI).

In addition to testing the core functionality, Wrisk will be expected to augment these tests with real life end to end scenarios, both simple and complex. Testing based on risk analysis is also recommended.

To avoid a formal release process, functionality will sometimes be deployed to Wrisk UAT environment on its own UAT test environment before testing has been completed by Hug team. Although this may result in duplication of bugs, it will provide us with early feedback and maintain a high velocity.

All bugs identified by Wrisk should be recorded in **Trello** with the steps required to replicate and any appropriate screenshots. These bugs will be updated by Hug, and we will send Wrisk notification when the functionality is available for retesting. The Hug team will identify any duplicates and update their status. Only Wrisk will be able to close Trello bugs.

## Automation Strategy

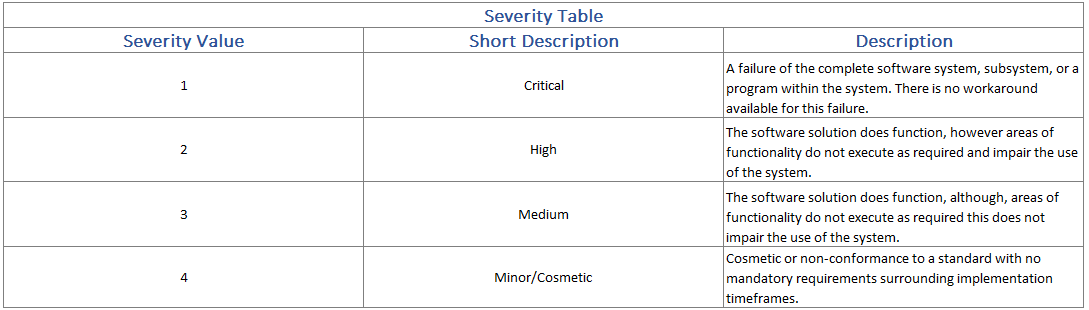
It is the aim of the team to utilise automation testing for regression testing wherever possible. All changes to the product should be manually tested to verify and validate the implementation before being automated. A decision should then be made as to whether the test can be automated or not, and if it adds benefit to the regression suite.

Testing of acceptance and processing of these events can be automated (marked in ‘Automation Scope’ column in the table).

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature** | **Description** | **Automation Scope** |
| F001 | Wrisk Motor Journey | Motor Journey in Agent View and Client View | In Scope |
| F002 | Policy Cancellation | Cancellation in Agent View and Client View | In Scope |
| F003 | MTA | MTA in Agent View and Client View | In Scope |
| F004 | Policy addons | Policy Addons (Legal expenses) in Agent View and Client View | In Scope |
| F005 | Search Policy details | Verify the search results in AV/Client view | In Scope |

## Bug Fixing

Bugs that are identified within a Sprint are documented and reported in VSTS. Any bugs that are identified of medium to minor severity should look to be addressed in the next or future sprints. If a bug blocks any tests cases from execution or is deemed of critical or high priority the team will address in the current sprint.



## Testing Principles

The following key testing principles shall apply to Wrisk testing.

* Both positive path and negative path testing will be carried out as per the scenarios identified in the table below

|  |  |  |
| --- | --- | --- |
| Sl.No | Scenario | Type of Test |
| 1 | Successful Wrisk Motor Journey for Av/Client | Positive |
| 2 | Successfully retrieve all information from Vehicle registration look up | Positive |
| 3 | Not retrieve any information from vehicle registration look up | Negative |
| 4 | Successful Stripe Payment | Positive |
| 5 | Document Generation | Negative |
| 6 | Email Notifications | Positive |
| 7 | Validate rating engine | Positive |
| 8 | Successful cancellation of policy via Agent/Client | Positive |
| 9 | Successfully do MTA via Agent/Client | Positive |
| 10 | Successfully renew a policy via Agent/Client | Positive |
| 11 | Successfully do 7 day policy via Agent/Client | Positive |
| 12 | Successfully add non insurer options in policy via Agent/Client | Positive |
| 13 | Cancel non insurer options from policy via Agent/Client | Positive |
| 14 | MTA for non-insurer options via Agent/Client | Positive |

* A risk-based testing approach will be taken for the Wrisk project. High priority items and their justification is given below

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Priority | Justification | Action |
| Rating engine | 1 |  | Focused testing on this component |
| Stripe Payment | 1 |  | To be covered in all phases of testing |

## Scope

Test scope shall be defined at the release/sprint test plan level which shall be aligned with the planned sprint activates.

## Sprint Test Planning

A brief test plan document shall be created for each sprint. This test plan may contain the following:

* Sprint Objectives.
* Scope of Testing.
* Assumptions and Risk.

## Sprint Test Design

Test Design outlines what must be tested as well as determining relative priority for each stage of testing throughout a sprint, referencing back to the phases outlined in the [Test Strategy](#_Test_Strategy) section.

Tests shall be prioritised based upon the risk of a failure occurring within a production environment, and the potential impact to the business should a failure occur.

Test case design should utilise techniques such as Boundary Value Analysis, Equivalence Partitioning, Decision Table Testing, State Transition Diagrams, Use Case Testing and Error Guessing. This is not an exhaustive list, and new techniques should be considered when designing test cases.

Test cases should be written in a manner that any tester with limited knowledge of the solution should be able to execute the tests.

Test cases shall be created within Team Foundation Server (TFS), utilising either the project web portal or Microsoft Test Manager.

## Sprint Test Execution

Executing test cases against the designated software during the sprint, exercises the solution to ensure that it meets the expected requirements. During the execution phases test cases steps shall be followed, observing the software’s behaviour and capturing any defects that are identified as a result.

Any defects found shall be logged against the software ensuring that they are categorised, assessed and ensure that the correct severity is assigned during each sprint. Associated re-testing once a fix has been implemented and the level of regression testing shall be determined by root cause and impact assessment performed by the sprint team.

Test execution shall occur as early in the sprint as possible to allow early feedback on implemented work items.

## Sprint Test Reporting

As part of the sprint completion activities a sprint test report shall be produced. This shall include the following information:

* Summary of Test Execution (Tests planned, tests executed, tests passed, tests failed, tests blocked or not tested).
* Details surrounding tests that were blocked or not tested.
* Work Items Tested in the Sprint and their status.
  + Any associated defects that are still open against these Work Items.
* Any new defects that have been raised and are still open.
* Any other information that is deemed relevant.

## Entry and Exit Criteria

Entry and Exit Criteria define when testing shall commence and complete as part of a sprint. The goal of each sprint shall be to make available a releasable product.

### Entry Criteria

The following criteria should be met to commence testing.

* Sprint work items have been agreed and are committed to by the team.
* Sprint test planning has been completed.
* Sprint test cases have been created.
* Initial sprint build has been completed by the team in readiness to test.
* All supporting infrastructure requirements have been updated and smoke tested.

### Exit Criteria

The following criteria should be met to complete testing.

* All committed sprint work items have been verified.
* All planned test cases have been executed.
  + Except for those that have been removed.
* All critical severity issues are verified as resolved.
* Sprint Test Report has been created.

### Sprint Abandonment

In some instances, a sprint may be abandoned due to impact through issues and changes in priority. When this occurs the Scrum Master shall notify the stakeholders. Where possible and necessary sprints can be extended or work items carried over. Abandoning a sprint is a last resort and attempts to mitigate impact should always be considered.

# Defect Management

Defects found throughout the products lifecycle shall be logged within Microsoft Visual Studio Team Services platform project.

## Defect Management Strategy – In Sprint

* Defect Identified & Replicated.
* Defect Logged in VSTS project.
* Defect Severity identified.
* Defect of Critical / High Severity sent to developer for resolution.
* Defect resolved by developer.
* Build with resolution provided.
* Original owner retests and verifies defect resolution.
* Defect either closed or sent back for rework.

# Artefacts & Deliverables

The following outlines are the sprint deliverables.

## Sprint

The following will be provided/Updated at the end of each sprint

* Wrisk Test Plan: This will be updated and amended to show any changes in spec or testing method that are required as the work continues (revisions will be noted and new versions will be created to provide traceability)
* Test Summary Report from VSTS: These will be generated at the end of each sprint and sent out by the test lead
* Automation Test Report there may be several of these reports produces during a single sprint but the latest version of all of these (for any products that have been worked on) will be sent out.
* Sprint QA Summary Report
* Product Release, including release notes, dependencies and deployment guidelines.

# Team Organisation

|  |  |  |
| --- | --- | --- |
| **Role** | **Name** | **Responsibility** |
| Scrum Master | Slava Solovei | Responsible for leading the scrum team in agile disciplines and providing sprint updates. |
| Engineer | William Kouda | Responsible for developing HH applications. |
| Engineer | Monica Tudor | Responsible for developing HH applications. |
| Engineer | Tim Knight | Responsible for developing HH applications. |
| Engineer | Tristan Gaydon | Responsible for developing HH applications. |
| Test Engineer | Tuely Saseendran | Responsible for testing of applications produced by HH scrum team. |