Contents

[Contents 1](#_Toc52877824)

[Document Information 1](#_Toc52877825)

[Document Distribution List 1](#_Toc52877826)

[Document Sign Off 2](#_Toc52877827)

[1. Background 3](#_Toc52877828)

[2. Test Approach 3](#_Toc52877829)

[2.1 Unit Testing 3](#_Toc52877830)

[2.2 Component Testing 3](#_Toc52877831)

[2.3 Exploratory Testing 4](#_Toc52877832)

[2.4 System Integration Testing 4](#_Toc52877833)

[2.5 Non-functional Testing 4](#_Toc52877834)

[3. Test Data Approach 5](#_Toc52877835)

[4. In Scope 6](#_Toc52877836)

[5. Out of Scope 6](#_Toc52877837)

[6. Risks & Issues 7](#_Toc52877838)

|  |
| --- |
| Document Information |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VERSION CONTROL** | | | | |
| **Document Location:** | | | Add link to sharepoint location here once uploaded | |
| **VERSION** | **STATUS** | **DATE** | **AUTHOR** | **COMMENTS** |
| 0.1 | Draft | 1-10-2020 | Greg Macdonald | Initial draft |
| 0.2 | Draft | 5-10-2020 | Greg Macdonald | Updated after initial peer review |
| 0.3 | Draft | 7-10-2020 | Greg Macdonald | Updated after new information shared regarding signature fields. |
|  |  |  |  |  |

|  |
| --- |
| Document Distribution List |

|  |  |
| --- | --- |
| **NAME** | **PROJECT ROLE / POSITION** |
| **Mark Whitaker** | Solutions Delivery Manager |
| **Frakhanda Shah** | Scrum Master |
| **Alison Trafford** | Systems Analyst |
| **Amerjit Singh** | Application Developer |
| **Manav Shah** | Application Developer |
| **Amin Abbas** | DevOps Technical Specialist |
| **Zabair Khan** | Application Support Specialist |
| **Greg Macdonald** | Senior Test Analyst |
| **Kamran Fiaz** | Senior Test Analyst |

|  |
| --- |
| Document Sign Off |

|  |  |
| --- | --- |
| **NAME** | **PROJECT ROLE / POSITION** |
| **Mark Whitaker** | Solutions Delivery Manager |
| **Frakhanda Shah** | Scrum Master |
| **Alison Trafford** | Systems Analyst |

# Background

The objectives of the project are like-for-like replacement of the three “processing routes“ currently provided for card payments within the provided deadline for valid licence of BizTalk. These processing routes being: Payment Notification (Valid and Invalid), Take Card Payments and Apply Payments.

The scope of the project is limited to a replacement of the current Card Payment function of the BizTalk servers and does not extend into replacing support systems such as CPM Web Proxy or CPM.

As an additional benefit, monitoring and logging for support and introducing alerts for failure reporting will be improved

# Test Approach

The objective of test activities is to ensure that new features satisfy business requirements with a focus on early defect detection in line with the “shift left” mindset. The aim is to adopt an “automation first” approach favouring quick, low-level unit tests. This differs to the approach normally taken by Solutions Delivery squads, which relies heavily on end-to-end testing of deployed software and involves more manual testing.

Leveraging automation at low-level will enable the squad to deliver changes more rapidly and with a consistent quality. Only exploratory testing, or tests which are deemed too onerous or unreliable to automate will be executed manually. Testing will involve demonstrating that the functional requirements have been satisfied and that no issues have been introduced to existing functionality. Regression testing will be carried out as and when required within each of the test levels when changes have been made to existing code. The following functional test levels will be included: Unit Testing, Integration/System Testing and End to End testing.

## 2.1 Unit Testing

Unit testing will be performed on the majority of developed code (generated code will be tested indirectly). Each class will have a corresponding test class. Each method in the class will have one or more unit tests written in the tests class to test logic flows. Tests will be written in MSTest framework and integrated in the Build pipeline in Azure DevOps. Class Dependency Injection will be used to mock (via MOQ) out any calls to other classes/APIs etc, so that isolation of the unit under test is achieved.

## 2.2 Component Testing

Component Testing will combine multiple classes/units that were tested in Unit Testing to create and test a complete component (API). Test cases will be documented in Azure DevOps and will undergo a peer review.

Functionality that is isolated to the component (API), such as request validation, field validation and component business logic will be tested - and would not need to be duplicated in System Integration Testing further down the line.

Where practical, real services will be tested against in this phase of testing. Calls to mocked out components (using MOQ) will only be used if the actual component is not readily available.

Tests will be written in MSTest framework, enabling them to be integrated in the Build pipeline in Azure DevOps**.** Evidence will be stored in a .txt file in DevOps for each test, which can be downloaded and viewed if required. Any failures in the test run will make the build fail and would have to be fixed in order to get release artefacts.

## 2.3 Exploratory Testing

Exploratory (aka unscripted) testing will be used in order to quickly identify additional testing scenarios that can be added to the other test phases. Insomnia/SoapUItoolwill be used to enter various data not already used in the scripted tests. Any failures would result in a bug being raised and a new test added to the scripted tests. The data used in each test is noted, to allow replication and to avoid unwanted repetition.

## 2.4 System Integration Testing

System Integration Testing will be carried out by manually testing the processes that trigger the 3 processing routes. This will be done using SOAP UI Pro/Insomnia.

Test cases will be documented in Azure DevOps and will undergo a peer review, these will be manual tests so test evidence will be taken at the time the test is run by the tester.

## 2.5 Non-functional Testing

Non-functional testing of the APIs will be carried out during the API testing

Application Performance Efficiency and Capacity:

The numbers of concurrent requests that we should use to Load, and Performance test the API’s should match what is currently hitting the existing services in production. JMeter, a third party tool, will be used to test performance of the new API’s.

Reliability:

Servers will be turned off at various points in the process to simulate unavailability. The expectation is that the messages will be stored at the last active part of the process and will then flow as normal once service is restored.

Audit and Robustness:

Logs on EventHub and Application Insightswill be checked by Development and Application Support teams respectively.

Compatibility and Integrity:

These will be proven as part of functional testing.

# Test Data Approach

Test data will either be taken from existing transactions found in logs or generated from the Verifone payments page and then reused and amended as required.

However, the messages contain a hashed ‘signature’ field (for data integrity, proof of origin?), so any changes to the original message would require a different value in the signature field. Therefore, a tool created by the Development team will be used to create the required signature field for the amended message.

Verifone payments page input data is limited a to small set of predefined data which can be found here:

<https://providentfinancialgroup.visualstudio.com/CCD_Solutions/_wiki/wikis/CCD_Solutions.wiki/1373/HC-Online-Payments-(HCOP)>

There is a single valid card number and CSV. Accepted/Declined transactions are determined by the payment amount (accepted – any amount not ending in 2, 5 or 7p, declined – any amount ending in 5p).

**3.1 Invalid Notifications and Schemas**

Verifone had not been known to have created any invalid notifications to date. We can therefore not use the Verifone payments page to generate such data. Therefore, the data will be handcrafted from the schema, example shown below. Similarly, notifications with invalid schemas will be handcrafted from copies of the valid ones.

****

# In Scope

* Functional testing of:
* Payment Notification Inbound and Outbound messages
* Take Card Payment Inbound and Outbound messages
* Apply Card Payment Inbound and Outbound messages
* Focus Database updates
* Non-functional requirements as mentioned in section 2.5

# Out of Scope

* Verifone web pages functionality
* Security testing
* Maintainability testing
* Back-up and Recovery testing
* Availability testing
* Accessibility testing

# Risks & Issues

To be considered in conjunction with project risks.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **#** | **Type** | **Description** | **Impact** | **Likelihood** | **Risk Profile** | **Comment** |
| **1** | Risk | As with any new approach, there is a risk of issues arising from insufficient skillsets, technical issues, unexpected circumstances or impediments caused by the implementation and adjustment to new methods of working. The issues could manifest themselves as undiscovered bugs and/or delays to delivery. | M | L | L | The new approach implements the “shift-left” concept embraced by the department and utilises test automation as the core of the test effort in line with the trends of the software development industry. While automation requires upfront effort to put the tests into place, it enables the features to be delivered incrementally with fast and efficient regression test runs, making it a suitable approach for agile, sprint-based project delivery. |
| **2** | Issue | In order to test resilience, we will turn off servers at various points in the process to simulate unavailability. It is not yet known how practical this will be . | M | L | L | This is an issue to be discussed with the Development and Platform teams to understand the practicalities of achieving this |
| **3** | Risk | There is only a limited experience of performance test too JMeter in the team. This could slow down the writing of the performance tests and/or limit the scope of the tests | M | M | M | There is expertise elsewhere in the Delivery Team, although there is the risk that access to that expertise may be restricted by other demands on their time |
| **4** | Risk | Only 2 people at a time can be logged onto the remote servers for checking the CPM Payment and Focus Agreement\_ Transaction tables have been updated correctly. | M | L | L | This has the potential to hinder testing progress if people from other projects are accessing these servers. |