# 

Statement of Work

Development of Customer Order Visibility Platform

Prepared for

Shell

Prepared by

**Yunus Emre Alpozen**, Apps Architect

Contributors

**Sanjay Mahato**, Pursuit Lead

**Shelley Campbell,** Account Delivery Executive

**Subhajit Chatterjee**,Apps Architect

Date: June 27, 2019

Version: 1.4

Table of contents

[Introduction 1](#_Toc12522446)

[1 Project objectives and scope 2](#_Toc12522447)

[1.1 Objectives 2](#_Toc12522448)

[1.2 Areas in scope 2](#_Toc12522449)

[1.3 Areas out of scope 16](#_Toc12522450)

[2 Project approach, timeline, and deliverable acceptance 18](#_Toc12522451)

[2.1 Scrum approach 18](#_Toc12522452)

[2.2 Sprint Principles 28](#_Toc12522453)

[2.3 Timeline 29](#_Toc12522454)

[2.4 Project governance 29](#_Toc12522455)

[2.5 Project completion 33](#_Toc12522456)

[3 Project organization 33](#_Toc12522457)

[3.1 Project roles and responsibilities 33](#_Toc12522458)

[4 Customer responsibilities and project assumptions 38](#_Toc12522459)

[4.1 Customer responsibilities 38](#_Toc12522460)

[4.2 Project assumptions 39](#_Toc12522461)

This Statement of Work (SOW) and any exhibits, appendices, schedules, and attachments to it are made pursuant to Work Order **6NLD198-242563-285503** and describes the work to be performed (services) by Microsoft (“us,” “we”) for Shell (“Customer,” “you,” “your”) relating to the Customer Order Visibility (project).

This SOW and the associated Work Order expire 30 days after their publication date, unless signed by both parties or formally extended in writing by Microsoft.

Introduction

Shell delivers goods, predominantly lubricants, and peripherals, to their business partners which are ordered through Shell Market-Hub and other channels. Business partner can be a car dealer or an industry customer using the products to maintain assets, such as fleet vehicles or production lines. Shell’s end customers have made requests to improve the ordering process – specifically to improve transparency of delivery, and the communication process. Through this project, Shell plans to provide a better visibility of progress of deliveries as packages pass from Shell warehouses to logistics partners and to bring tracking capabilities to the customer enabling them to receive notifications of deliveries. In order to achieve this one of the key insights will need to come by integrating with a Transport Management Service (TMS) or 3PL (3rd Party Logistics) provider for which a limited trial was implemented focusing on deliveries in Turkey. By considering the learnings from this initiative, this project will deliver a minimum viable product (MVP), of defined scope, which establishes a strategic capability, enabling Shell to scale this integration globally.

The development of the Customer Order Visibility will be characterized by:

* Agile/SCRUM development and DevOps principles
* Continuous Integration and Continuous Delivery (CI/CD)
* Implementing the initial base foundation and develop a limited scenario to aggregate the data and the view on a customer order
* Work with Shell to do a handover of the development to Shell to take it up further

This project is proposed on a fixed fee, fixed capacity basis, to deliver the intended MVP scope in defined functional blocks. User stories will be decomposed in Sprint 0, and acceptance criteria agreed, at the end of which scope will be fully locked and the remaining Sprints planned. Some risk is acknowledged with this approach. Should any major deviation occur through current unknowns at the end of Sprint 0, or due to major change in scope expressed by the Shell Product Owner, a formal governance process (to be agreed) will notify client of impact and consequence so that scope can be reprioritized to fit planned capacity for delivery.

# Project objectives and scope

## Objectives

The primary goal of this engagement is to provide Shell with an application development capacity to build the base foundation of Customer Order Visibility using Microsoft recommended practices while leveraging the benefits of Microsoft Azure PaaS and Azure DevOps services. The outcome will deliver a releasable minimum viable product (MVP) which enables Shell to replace its current tactical (Turkey) prototype.

The objective is to provide a DevOps environment of continuous learning, adopting recommended practices as they appear, with a smooth integration of the development team with the customer teams consisting of Shell and (where applicable) its third party resources. However, for this contract, the list of outcomes will be limited by the prioritized scope that can be delivered within the timeline of 16 weeks by the cross-functional Microsoft team.

Microsoft development team will support with designing the architecture while exploiting available Azure PaaS components with a view to the component roadmap to maximise the agility of the release, within the short time period proposed. This project would enable for Shell increased stability, performance, and manageability compared to the tactical (Turkey) prototype.

Shell is solely responsible for the Product Roadmap and the generation, prioritization and sign-off of Epics, User Stories, and Acceptance Criteria which determine the Features and Functional product backlog items (PBI’s). Shell should also take accountability for defining and sharing business testing use cases. Microsoft will support Shell with estimating the size of User Stories, confirming readiness to commit (definition of ready), Sprint Planning the generation of product-backlog items (PBIs) to improve the architecture (Technical PBIs), including data management, especially where these relate to the use of new or changed Azure services.

## Areas in scope

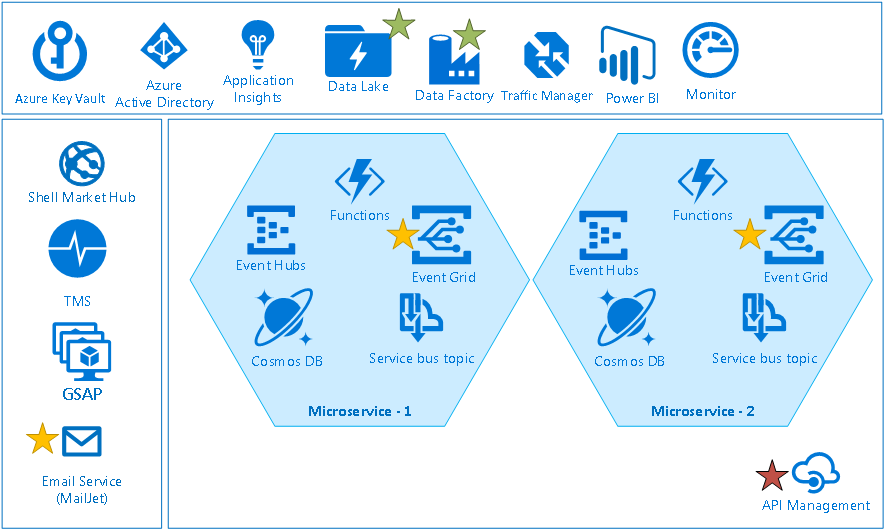
### General project scope

Microsoft will provide Shell with development teams staffed as defined in the Project Organization section. The development teams will follow recommended agile practices (described further in the Scrum approach section) as Microsoft builds the minimum viable product for the Customer Order Visibility Platform.

Microsoft will provide services in support of the scope, provided below. Final scope will be validated, aligned with capacity and baselined following completion of Sprint 0

| Area | Epic | Description |
| --- | --- | --- |
| Architecture | Design “to be” architecture | Microsoft will help to define target architecture based on serverless microservices technologies available in Azure, as far as this architecture does not yet exist. On a continuous basis, this architecture will be improved to use newly available Azure PaaS services and complementing third party products and services to cover the full set of functional and technical requirements of the Customer Order Visibility Platform. |
|  | Build design principles | Define coding standards and design principles |
|  | Technical Quality Assurance | Provide oversight and govern technical quality by code review and architecture reviews on a regular basis. Solution architecture design will be kept improving as new requirements are introduced. |
|  | Plan Coexistence | Plan coexistence with current implementation and plan integration points. |
| Integration | Shell Market-Hub | Integration with Shell Market-Hub will be limited to exposing APIs which can be consumable by Shell Market-Hub to monitor customer orders.  Shell Market-Hub implementation is out of scope for this engagement. |
|  | TMS | This activity will be limited to exposing an API which can be consumable by Transporean to push delivery update messages.  Shell is solely responsible to manage communication and align timelines for deliverables along with the plan proposed by Microsoft |
| Core Foundation | API Publishing | Providing a public endpoint for Transporean to push order update message for a delivery. This activity covers storing raw data on a Cosmos DB for further processing and raising events on status changes. |
|  | Authentication & Authorization | Configuration of API Management endpoints to allow subscription-based authentication and authorization.  Configuration of APIGEE will be carried out by Shell. All exposed APIs will be protected by Azure API Management. APIGEE may introduce an additional layer of authentication and authorization. This decision will be taken during Sprint – 0 and design activities. |
|  | Monitoring & Telemetry | Configuration of Application Insights and Azure Monitor services to capture usage and exceptions. |
| Capturing Delivery Messages | Accepting all valid messages | Any message sent by transport management system and/or 3rd party logistics will be validated only on message format which is limited to having a proper JSON format and having required sections which will be agreed during Sprint – 0 along with Shell Architects.  All valid messages will be pushed to an Azure Service Bus topic (Valid-Raw-Messages) to be processed further. After format validation and placing messages on the Azure Service Bus, a success response will be returned to caller.  All invalid messages will be pushed to another instance of Azure Service Bus topic (Invalid-Raw-Messages) to be processed further. For invalid messages, a specific failure messages will be returned to the caller to notify caller.  Validating messages with data format and/or applying business rules will be out of scope for this activity.  It is assumed each message will be less than 2KB in order to achieve desired performance targets. |
|  | Storing raw messages in a Cosmos DB | Whenever a message with a valid format is sent to Azure Service Bus Topic (Valid-Raw-Messages), a new Azure Function instance will be triggered to store the raw message on a Cosmos DB instance.  During this activity, partitioning strategy for this Cosmos DB instance will be designed. Stored data can be captured using another tool such as Azure Data Factory to be stored in a Data Lake to analysis. This activity will be out of scope for this engagement. It is under Shell responsibility to carry out this data movement. |
|  | Identifying events from messages | Whenever a message with a valid format is sent to Azure Service Bus Topic (Valid-Raw-Messages), a new Azure Function instance will be triggered to process message. Processing will be limited to following activities:   * Identifying correct shipment details * Identifying customer details * Identifying the event * Validating data types. * All valid messages will be sent to another Azure Service Bus Topic (Valid-Processed-Messages) * All invalid messages will be sent to another Azure Service Bus Topic (Invalid-Processed-Messages) |
|  | Storing Identified events from messages for a single delivery | Whenever a valid message is sent to Azure Service Bus Topic (Valid-Processed-Messages), an Azure Function instance will be triggered to store the message on a Cosmos DB database.  During this activity, partitioning strategy for this Cosmos DB instance will be designed.  This database will be the single truth for querying delivery statuses.  Stored data can be captured using another tool such as Azure Data Factory to be stored in a Data Lake to analysis. This activity will be out of scope for this engagement. It is under Shell responsibility to carry out this data movement. |
|  | Storing identified events from messages for a single customer | Whenever a valid message is sent to Azure Service Bus Topic (Valid-Processed-Messages), an Azure Function instance will be triggered to store the message on a Cosmos DB database.  During this activity, partitioning strategy for this Cosmos DB instance will be designed. This database is designated to act as a materialized view to speed up queries. Any time this database can be deleted and can be reconstructed from original data stored for each delivery.  Stored data can be captured using another tool such as Azure Data Factory to be stored in a Data Lake to analysis. This activity will be out of scope for this engagement. It is under Shell responsibility to carry out this data movement. |
|  | Detecting notification from identified events | Whenever a valid message is sent to Azure Service Bus Topic (Valid-Processed-Messages) an Azure Function instance will be triggered to check if there is a need to send a notification to customer based on customer notification preferences. If is necessary to send a notification, a message will be pushed to another Azure Service Bus Topic (Customer-Notification). If there is no need to send a notification, no action will be taken. |
| Notification Management | Checking notification | Whenever a message is sent to Azure Service Bus Topic (Customer-Notification), an Azure Function instance will be triggered to check customer preferences and send email messages to predefined email addresses.  Based on customer locale and notification type relevant email template will be used. Email templates will be stored as HTML files on a blob storage. Email templates will contain some tokens to be replaced to generate content.  Actual generated content will be sent as a message to an Azure Service Bus Topic (Customer-Notification-Content). A different message will be placed if there are multiple recipients or multiple notification channel is requested for that message.  A flag will be stored on a Cosmos DB to ensure that customer is not receiving multiple notifications for the same event.  Email templates will not be moderated.  Email templates will be provided by Shell. Microsoft responsibility is limited to upload templates to a blob storage and use them. |
|  | Customer notification preferences | A REST based API will be published for Shell Market-Hub to gather and modify customer notification preferences. Customer notification preferences will contain following information:   * An array of events: * Each Event will have following information:   + Notification Types (Only email will be available)   + Trigger: Time or Distance   + Units: Hours or Days for Time, Kilometers or Miles for Distance   + Threshold: Numeric Value * Notification Address(es): can be multiple * Notification Locale   Customer notification preferences will be stored on a Cosmos DB.  Standard CRUD Operations (Add, Update, Delete, Get) will be implemented in the scope of this engagement. Get will be limited to query only for a single customer. Listing all customers will be out of scope.  An Azure Function instance will be developed for each CRUD operation.  Migrating existing customer notification preferences will be out of scope. |
|  | Sending email notifications | Whenever a message is sent to Azure Service Bus Topic (Customer-Notification-Content), a new instance of Azure Function will be triggered.  MailJet will be used to send email message. It is assumed Shell will provide access credentials for MailJet;  **Note:** MailJet configuration is out of scope for this engagement. |
|  | Storing notification messages | Whenever a message is pushed to Azure Service Bus Topic (Customer-Notification-Content), the content of notification will be stored on a Cosmos DB for auditing purposes.  Displaying and/or querying data stored in this database will be out of scope. This database will be used as a read-only data store. |
| Query Delivery Status | Querying Order Status for a delivery | A REST based API will be published for Shell Market-Hub to gather delivery status for a specific order. This will be published in APIM.  For this API, an Azure Function will be implemented to query data stored in Cosmos DB database for the relevant data.  Only single REST based API will be developed. It will return entire data stored in database. Filtering of returned data should be performed by the caller, if necessary. |
|  | Querying all orders for a customer | A REST based API will be published for Shell Market-Hub to gather all deliveries for a specific customer.  An Azure Function will be implemented to query data stored in Cosmos DB database for relevant data.  Migrating existing orders is out of scope for this engagement. |
| Error Handling | Error handling | Building a reliable solution with a configurable retry mechanism for transient failures and data format errors. Whenever a message is sent to Azure Service Bus Topics (Invalid-Processed-Messages or Invalid-Raw-Messages) a new instance of Azure Function will be triggered to store these messages.  Messages which cannot be processed due to a validation and/or format error will be captured separately on a Cosmos DB database. No further action will be performed on these captured messages. |
| Testing | Unit Testing | A unit test will be developed for each code piece developed by Microsoft. This activity will be carried out by developers by mocking any external interface (including but not limited to Azure services). Unit tests will be executed within CI/CD pipelines and will be designed to be executed in offline mode. |
|  | Test Automation | This activity will be limited to executing happy paths only and one failure scenario at most (wherever it is applicable).  Test automations will be executed daily and will be part of nightly builds.  Test results will be published to Azure DevOps as an evidence. |
|  | Performance Test | This activity will be limited to only REST based API endpoints.  Only during final stabilization sprint, this testing will be carried out through APIGEE. In earlier sprints, APIGEE will be bypassed.  Up to 10 APIs will be tested in scope of this engagement.  Performance testing will be carried out with 50 requests per second and no endurance test will be executed. |
| Documentation | High Level Solution Design | A document will be crafted during Sprint – 0 to define scope items and design decisions.  Word based Microsoft documentation format will be used. |
|  | Detailed Level Solution Design | A document will be crafted during delivery sprints to articulate implementation details.  Documentation will be published on Azure DevOps Wiki pages. |
|  | Test Reports | Test results will be published to Azure DevOps. |
|  | Azure DevOps configuration and deployment guide | Azure DevOps configuration will be defined as a Wiki Page on Azure DevOps to ease hand over to Shell teams.  Deployment guide will contain necessary configuration required to deploy the solution to the production environment. Deployment guide will be published as a Wiki page on Azure DevOps. |
| Security | Secure Development Lifecycle | Microsoft recommended practices will be followed during delivery of this engagement to achieve desirable security quality to the extent possible with available capacity. Microsoft will provide transparency through Azure DevOps tool if there are any known security bugs and mutually agree on the resolution.  Any connection string or credential will be stored in an Azure Key Vault instance. Accessing Azure Key Vault data will be achieved using Managed Identity. It is Shell’s responsibility to ensure Managed Identity accounts are created with sufficient permission on Azure Active Directory. |
| Azure DevOps configuration | Azure DevOps Configuration | Azure DevOps configuration will be performed by Microsoft development teams to make certain utmost benefits are leveraged and Microsoft practices are followed.  Shell SonarQube will be integrated to existing pipeline with Shell’s support. |
|  | Production Deployment | Deployment to production environment and fulfilling the activities required by Shell processes will be under responsibility of Shell teams. |

### Solution Components and indicative High-level Solution Architecture



The solution components indicated with a green star in the illustration above signify that Shell agrees to utilize them for the implementation. Those that are in yellow indicate they are not available in Shell Azure environment and an exemption process is to be followed by Shell to make these available for this project. Those in red indicate they are not allowed for the implementation. Shell should follow their exemption process to make these components available for this project.

In figure given above, only high-level solution components are listed:

* Traffic Manager
* Frontend Web Application
* Functions developed using Microsoft .NET Core 2.0
* Service Bus
* Other PaaS Components

The following diagram shows the indicative solution design. It is subject to revision based on the outcome of the Sprint – 0 workshop.



Figure 1 - Shell Market-Hub Queries Status of a delivery



Figure 2 - Shell Market-Hub updates customer delivery preferences



Figure 3 - Entire flow when a new message is arrived from TMS

### Software products and technologies

The products and technology that are listed in the following table are required for the project with the information that is provided today. The Customer is responsible for obtaining all identified licenses and products (including future ones from third party providers if applicable).

| Product and technology item | Version | Ready by |
| --- | --- | --- |
| Microsoft Azure DevOps Services | Current | Start of project |
| Microsoft Azure subscription | Not applicable | Start of project |

### Environments

All environments used for integration, preproduction, and production use of the developed software, supporting systems, and development lifecycle will be supplied and maintained by the Customer.

The Customer will provide an Azure subscription for all environments listed in the below table. The Customer will also provide Microsoft with administrative control to build the development, test and UAT environments. The Microsoft team will be responsible for deploying & managing builds on the development and test environments.

Additional environments might be required to support parallel development & test teams. These will be determined in Sprint 0. We would evaluate the need for additional environments during the course of the project (especially for parallel releases, etc.) and make appropriate requests to Shell.

The following proposed environments will be required to deliver the project (to be aligned and agreed upon during Sprint 0).

| Environment | Location | Responsible for preparation | Ready by |
| --- | --- | --- | --- |
| Development | Azure | Microsoft | Within 3 week of project start |
| Test | Azure | Microsoft | Within 3 weeks of project start |
| Automation | Azure | Microsoft | Within 3 weeks of project start |
| Performance | Azure | Microsoft | Within the 1st week of Sprint 1 |
| User acceptance testing (UAT) | Azure | Microsoft | Within the 1st week of Sprint 1 |
| Preproduction/Demo | Azure | To be determined during Sprint 0 | Within the 1st week of Sprint 1 |
| Production | Azure | To be determined during Sprint 0 | Within Sprint 1 |

### Testing and defect remediation

#### Testing

The following testing activities would be executed as part of the engagement.

| Test type (environment) | Description | Responsibility | | |
| --- | --- | --- | --- | --- |
| Has responsibility for testing? | Provides data and test cases | Provides guidance and support |
| Automated unit tests  (development) | Automated tests that cover a single component or element of code. | Microsoft | Microsoft | Microsoft |
| Automated integration tests  (Automation) | Automated tests that combine 2 or more system components. | Microsoft | Microsoft | Microsoft |
| Functional Testing (test) | As Microsoft is building an API based platform the functional testing will be limited to validation of data input and expected output based on the business rules as captured in the user stories and acceptance criteria. | Microsoft | Shell | Shell |
| Performance Testing (Performance) | Performance testing for the REST based API endpoints. | Microsoft | Shell | Shell |
| UAT  (UAT) | Tests the user functionality of key real-world scenarios. UAT will be conducted over the course of the project according to the UAT timeframes agreed upon during Sprint 0 (as described in the Sprint 0 section). Feedback from UAT (bugs or new user stories) and other backlog items will be prioritized in the product backlog. | Shell | Shell | Microsoft |

**Test Guidance and Support** is defined as:

* Answering technical questions during testing
* Providing informal knowledge transfer on the solution and business capability being tested
* Addressing issues and assist in registering bugs that are discovered during testing

During testing, Shell and Microsoft will work together on solution related issues and their priority. Shell and Microsoft will mutually agree to the initial priority for each defect. Any discordance within the joint Shell and Microsoft development teams regarding defect priority categorization will be managed through the defined Escalation process detailed in Section 2.4.4.

#### Defect remediation

If defects are identified during testing, the defect will be triaged and the priority of the item will be jointly agreed upon by the Customer and Microsoft. Defect prioritization is defined in the following table.

| Priority | Description | Remediation in scope? |
| --- | --- | --- |
| P1 | **Blocking defect**  Development, testing, or production launch cannot proceed until this type of defect is corrected. A defect of this type blocks further progress in this area. The solution cannot ship and the project team cannot achieve the next milestone until such a defect is corrected. | Yes, unless de-prioritized in the backlog by Shell’s product owner. |
| P2 | **Significant defect** This type of defect must be fixed prior to moving to production. Such a defect, however, will not affect test plan implementation. | Yes, unless de-prioritized in the backlog by Shell’s product owner. |
| P3 | **Important defect** It is important to correct this type of defect. However, it is possible to move forward into production using a workaround. | Only taken up based on available sprint capacity and planned by the Project team or through the change management process defined in 2.4.5  Otherwise, the same shall be added to the product backlog for future releases. |
| P4 | **Enhancements and low priority defects** P4 defects consist of feature enhancement and cosmetic defects. These include design requests that vary from original concepts. | Only taken up based on available sprint capacity and planned by the Project team or through the change management process defined in 2.4.5.  Otherwise, the same shall be added to the product backlog for future releases. |

Defects will be added to the product backlog as backlog items. Defects and other backlog items will be prioritized by the Customer product owner and added to future sprints.

If P1 or P2 defects are found in the UAT environment, the development teams will collaborate with the product owner to plan remediation and test in subsequent UAT code drops prior to release. The development teams will inform the product owner about the potential impact a defect fix could have on the current sprint. Some backlog items might need to be deferred from the current sprint to the product backlog in order to accommodate the work required to fix the defect. The development teams and the product owner will together determine what needs to be deferred from the current sprint, if anything.

The definition of done, agreed to in Sprint 0 (see the Sprint 0 section), should specify that a user story is not complete if there are any open P1 or P2 defects related to the story.

Before Final UAT, Pilot and Production deployment, for all identified P1 and P2 issues, resources will be allocated for successfully correcting the issues.

## Areas out of scope

Any area not explicitly included in the 1.2 – Areas in scope section is out of scope for Microsoft during this engagement. Areas out of scope for this engagement are listed in the following table.

| Area | Description | |
| --- | --- | --- |
| Product licenses and subscriptions | Product licenses (Microsoft or non-Microsoft) and cloud service subscriptions are not included. |
| Hardware | Microsoft will not provide hardware for this project. |
| Integration with third-party software | Microsoft will not be responsible for integration with third-party software which are not explicitly listed in scope section. |
| Data migration | Data migration from different sources are not in scope for this project, as far as these are not part of the solution design. Data migration through automated build and release process among environments is in scope. |
| System integration | System integration and interfaces other than listed in Section 1.2.1 are not listed explicitly in scope for this project. |
| Product bugs and upgrades | Product upgrades, bugs, and design change requests for Microsoft products are not in scope for this project. For items affecting delivery of the solution, Microsoft team will provide necessary input. However, issue resolution should be managed through Microsoft Premier Services. |
| Source code review | The Customer will not provide Microsoft with access to non-Microsoft source code or source code information. For any non-Microsoft code (including but not limited to open source), Microsoft services will be limited to the analysis of binary data, such as a process dump or network monitor trace. |
| Organizational change management | Designing—or redesigning—the Customer’s functional organization is not included. |
| Deployment, installation, configuration, and testing | The following items are not included:  Application deployment on client devices.  Installation and configuration of server hardware or network resources.  Installation, configuration, and testing of non-Microsoft software other than software identified as within scope.  Testing and configuration of applications and services outside of those required to support the deployment of the solution.  Troubleshooting or remediation of existing network and storage systems, as far as they are not part of the solution. |
| Testing | Testing and configuration of applications and services outside of those required to support the deployment of the solution are not in scope. |
| Endurance/Stress Testing | Endurance and/or stress testing will not be executed in scope of this engagement |
| Security / Vulnerability / Penetration Testing | Any kind of security testing including but not limited to penetration, vulnerability, security testing is not included. Executing these tests can be carried out by a 3rd party. Resolving any findings that are originated from Microsoft developed code will be triaged and planned. Not all security issues can be fixed. As an example, static code analysis if supported by Azure DevOps can be implemented as a step in the build and release pipeline. The design of build and release pipeline will be agreed in Sprint 0. |
| Notification Settings | * Moderating Email and/or SMS templates are out of scope * Sending SMS messages * Preparing the HTML Content for Email Templates * It is assumed Shell will provide access credentials for MailJet; MailJet configuration is out of scope for this engagement. |
| UI | No user interface will be provided in scope of this engagement. All activities will be limited to building API endpoints. |
| Integration | Shell Market-Hub implementation is out of scope for this engagement. | |
| Data Messages | Validating messages with data format and/or applying business rules will be out of scope for this activity. | |
| Capturing Delivery messages | Stored data can be captured using another tool such as Azure Data Factory to be stored in a Data Lake to analysis. This activity will be out of scope for this engagement. | |
| Customer notification preferences | Migrating existing customer notification preferences will be out of scope | |
| Storing notification messages | Displaying and/or querying data stored in this database will be out of scope. This database will be used as a read-only data store. | |
| Access and Identity | It is Shell’s responsibility to ensure Managed Identity accounts are created with sufficient permission on Azure Active Directory. | |
| Customer Orders | Migrating existing orders is out of scope for this engagement. | |
| Production Deployment | Deployment to production environment is out of scope for Microsoft. | |
| Deployment to In-Scope Environments | Microsoft development team will deploy the solution to two regions in Europe. Deploying solution to more regions and/or outside of Europe is out of scope. | |
| GDPR Compliance | Making solution GDPR complaint is out of scope. Solution will rely on GDPR compliance of individual Azure components and running only on Azure Europe regions. | |
| Hypercare | Hypercare is a post-production support which would be addressed via the change management process if requested by Shell. | |
| Moving Stored Data | Stored data can be captured using another tool such as Azure Data Factory to be stored in a Data Lake to analysis. This activity will be out of scope for this engagement. It is under Shell responsibility to carry out this data movement. | |
| Monitoring & Telemetry | KPI based monitoring | |
| Localization | Support for additional languages (other than English) is out-of-scope | |

# Project approach, timeline, and deliverable acceptance

## Scrum approach

### Sprint process

Microsoft will undertake an iterative development approach that is based on a fixed-duration, variable-scope process known as the scrum process (<http://scrumguides.org>). The key tenets are as follows:

* Joint ownership of decisions.
* Short implementation units (sprints).
* Prioritization of business objectives in a product backlog.
* Time-bound planning for each sprint.
* Emphasis on the remaining work.
* Sprints that produce a functional solution.
* Sprint demonstrations that are time-restricted and have regular checkpoints.
* Regular retrospective meetings that may be used for course correction.

Each of these activities will be later described in further detail and time-boxed to the noted total duration.

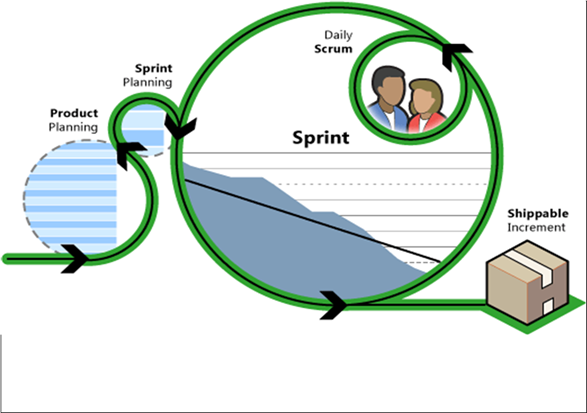


Figure 4: Agile Scrum approach

For each of the activities described in the following sections, the following Customer resources will attend, contribute, and be empowered to make decisions:

* Customer project manager.
* Customer technical decision makers, such as architects.
* Customer business decision makers, such as product owners.
* User representatives, such as users or their proxies.

If additional Customer resources are required, they will be specifically listed in the following sections that describe the workshops.

Followings are a list of common guidelines

* Requirements will be captured in the form of personas, user journeys, epics, features, product backlog items, user stories.
* Personas describe the roles of the people using the system.
* User journeys are simple descriptions of the work processes currently done by the personas, as well as desired future process changes. User journeys are optional and may be created during or after the rapid envisioning workshop to gain clarity on complex workflows.
* User stories are work items in the product backlog. They have the following elements, at a minimum:
  + A single persona who is the primary stakeholder in the outcome of the user story
  + A simple description of what the system should do
  + A simple description of the value of the result
  + A set of acceptance criteria that define when the user story is complete
  + Optionally, wireframe documents may also be created to supplement user stories if the application requires a high degree of user interface design.
* A feature is a higher-level, more abstract description of what the system should do. The sole purpose of features is to categorize user stories. Epics are more abstract than features. The sole purpose of epics is to categorize features. Features and epics are optional and may be used to help organize a large number of user stories.

### Engagement initiation

Before beginning the project, the following prerequisites must be completed.

| Category | Description |
| --- | --- |
| **Microsoft activities** The activities to be performed by Microsoft | Conduct a preinitiation call or meeting to initiate development team formation and communicate expectations.  Document the project launch prerequisites using input from this SOW.  Track the status of launch prerequisites and adjust the engagement initiation phase start date accordingly.  Conduct a detailed walk-through of the SOW with the Customer to agree upon an initial project schedule and approach.  Identify the Microsoft project team members for the Rapid Envisioning workshop and Sprint 0 phase. |
| **Customer activities** The activities to be performed by the Customer | Attend and participate in the preinitiation call.  Assign project initiation and launch prerequisites responsibilities to accountable Customer leadership and establish target completion dates.  Complete the project initiation and launch prerequisites.  Staff the project with the required Customer resources in the time frames agreed upon in the preinitiation call. |

### Sprint 0

Sprint 0 will be **4 weeks** in length. The below activities are indicative only, and Microsoft will deliver to the extent of available capacity or extend the Sprint 0 schedule as required on Microsoft’s discretion.

| Category | Description |
| --- | --- |
| **Microsoft activities** The activities to be performed by Microsoft | Review the scope and objectives.  Collaborate with the Customer to develop:   * + A problem statement.   + Vision statements.   + Personas.   + User journeys.   + Epics and features.   + User stories for the product backlog.   Create an initial list of non-functional requirements, such as performance, security and scalability needs.  Develop a recommended high-level technical architecture.  Help prepare the development environment.  Create initial code assets as an implementation of parts of the solution architecture.  Set up application lifecycle management (ALM) and DevOps that include building, releasing, and deploying a shippable increment.  Collaborate with the Customer on an estimated release plan based on the initial backlog of user stories.  Collaborate with the Customer product owner to create and elaborate proposed scope for Sprint 1 & 2, including a set of user stories that are ready for sizing, design, and development.  Provide coaching to help the product owner manage the product backlog.  Identify impediments to efficient development, including areas that require more elaboration, like proofs of concept or other architectural discovery tasks.  Collaborate with the Customer to create a definition of ready checklist that allows user story formulation. This checklist would drive completeness of user story formulation and allow Microsoft to accept a user story as “ready” for development.  Collaborate with the Customer to create a definition of done, that is, what constitutes completed user stories. That criteria will be used by the development teams to decide when a story is complete.  Collaborate with the Customer to determine how much information needs to be available before the development teams may work on user stories.  Define a test strategy and plan for all in-scope testing defined in the Testing and defect remediation section. If additional testing is determined as necessary during Sprint 0, it may be added following the change management process.  Collaborate with customer to define the number of users and scalability patterns with regards to Performance Testing.  Determine ALM and DevOps processes and tools.  Re-baseline the estimate of effort after detailing user stories during backlog grooming.  Agree upon UAT entry and exit criteria  Explore external dependencies.  Create a risk list.  Agree upon an estimation methodology with the Customer.  Establish project communication plan to manage project interdependencies with current development teams managed by Shell.  Align on lifecycle, build and release management with the current development teams.  Establish project management & governance processes, templates and tools with inputs from Shell  Create and review the project staffing plan   * + Agree composition of all development teams   + Agree Business Analysts (BAs) attachment to those teams * Collaborate with the customer to determine the activities and the effort needed to prepare for deployment of the MVP into the production environment and adhering to the Shell requirements for a production deployment. Should an agreement be reached, scope and effort related to Production deployment support would be planned and will be quantified during Sprint 0 and to be commercially agreed either via a Change Request or a new contract. |
| **Customer activities** The activities to be performed by the Customer | Attend and participate in the envisioning workshop sessions.  Help define user stories.  Provide updated background information, documentation, and business requirements.  Identify a product owner who is empowered to make business prioritization decisions and act as a single point of contact for requirements questions.  Identify Customer team members who will be available for the duration of the project.  Clarify requirements as needed.  Help Microsoft develop the technical architecture  Help prepare the development environment, where necessary.  Collaborate with Microsoft to create a proposed scope for Sprint 1.  Provide help removing any impediments.  Agree strategy and processes of Demos and POC environments  Baseline the project delivery plan – timeline, releases, release / Sprint plan, Sprint entry / exit criteria, deliverables acceptance criteria  Prepare scope and boundary of activities that current development team at Shell need to undertake and which should be clearly articulated for Microsoft development team as part of the transition and handover.  Data migration activities will be agreed in Sprint 0.  Collaborate with Microsoft to define the number of users and scalability patterns with regards to Performance Testing.  The design of build and release pipeline will be agreed in Sprint 0.  Share and setup Shell requirements and process for production deployment of the solution. |
| **Entry Criteria**  Allows the Sprint to begin | Project initiation complete with core team members on-boarded |
| **Exit Criteria**  Allows the Sprint to conclude before moving to the next Sprint / phase | Prioritized and updated product backlog right-sized to contracted capacity  Agreed High-level Assumptions around Product Backlog  Agreed Sprint plan and Sprint backlog for the development sprints  Agreed scope of business capability included in the first development release  Baseline solution architecture established  QA processes agreed  Agree on Definition of Ready and Definition of Done  Test plan reviewed  Solution unit and system test environment ready for use  Solution build & deployment plan (including code quality metrics) reviewed and approved  Agreement on modifications (if any) on current build and release pipeline  Agreed Acceptance Criteria for all Non- Functional Requirements documented as of the start of the project  Initial effort estimates by Feature as it relates to Sprint planning and releases  Scope of test case automation defined  Re-validate effort estimation at the end of Sprint 0. Should there be a variance to scope, capacity and/or timelines compared to the signed contract, then a Change request as described in Section 2.4.5 is to be initiated.  Finalize the UAT approach  Agreed UAT Entry and Exit Criteria |

### Development sprints

Following the rapid envisioning workshop and Sprint 0, the development process will be initiated through development sprints. For the project three (3) sprints are planned. Each sprint will last three (3) weeks.

Before sprint planning starts, the Customer product owner will collaborate with Microsoft to create a proposed sprint scope. This sprint scope will consist of a set of user stories that Microsoft and the product owner estimate may be completed during the sprint.

The Product Owner (or at his/her discretion his/her delegate) has responsibility to determine the order of the user stories in the backlog (taking into account Microsoft’s advice on technical dependencies), balancing highest business value and the level of effort required to complete against defined acceptance criteria of the user stories. The scope of the Sprint must be finalized by Shell Product Owner as per the Sprint Principles in Section 2.2. The Microsoft development teams will have activities/tasks to complete design, develop, test, build and deployment of the solution for Release Stabilize and UAT.

The first day of every sprint will be set aside for sprint planning for that sprint. The Microsoft development teams and the Customer product owner will attend. The Microsoft development teams would lead the meeting and the following activities will take place:

* Each user story will be reviewed by the Microsoft development team. Microsoft development teams will determine if there is sufficient information to begin development. They might seek clarification from the Customer product owner. If there is insufficient information to develop a story and the product owner cannot provide clarification during the meeting, the story may be deferred to a later sprint by the Microsoft development teams.
* The Microsoft development teams will determine which user stories may be accomplished during the sprint. If the proposed scope is too large, the development teams will collaborate with the Customer product owner to defer stories to a later sprint. If the proposed scope is too small, the teams will collaborate with the Customer product owner to add user stories. The user stories selected for the sprint are solely determined by the development teams.
* The Microsoft development teams will work together to decide how the work will be accomplished. This may include design discussions, updates to the architecture, and a breakdown of user stories into tasks.

During the sprint, the Microsoft development teams will build out the solution in accordance with the planned architecture. Daily standup meetings will be performed by the Microsoft development teams to keep everyone informed and to report any impediments.

During the sprint, if the Microsoft development teams determines that a backlog item cannot be completed within the sprint duration, it will be deferred to a later sprint after consultation with the team and the Customer product owner. If the Microsoft development teams have extra capacity in a sprint, the Microsoft development teams will collaborate with the product owner to select backlog items to be added to the sprint scope (within baselined project scope after Sprint 0). The Microsoft development teams are the sole decision maker on scope changes during the sprint.

The last day of the sprint is dedicated to demonstrating the functionality that has been achieved in the sprint and to carrying out a retrospective of the sprint. This is conducted in two parts.

* Sprint review: a sprint review meeting is held at the end of the sprint to inspect the increment and adapt the product backlog if needed. The product owner and Customer stakeholders will attend.
* Sprint retrospective: the sprint retrospective is an opportunity for the development teams to inspect themselves and determine if there are any improvements that need to be enacted during the next sprint.

| Category | Description |
| --- | --- |
| **Microsoft activities** The activities to be performed by Microsoft | Create and design, development and test tasks for Sprint backlog in scope for the current sprint  Execute assigned tasks to complete the work (design, development and test) and update task status in Azure DevOps  Review Azure DevOps task status and monitor Sprint progress  Create Sprint plan & backlog for the next development Sprint  Update test plan  Update solution backlog (backlog grooming resulting from various reasons including defects, progressive elaboration, scope change, issue resolution, risk mitigation, sprint backlog changes)  Conduct daily stand-up meetings  Participate in issue triage meetings, fix sprint issues (P1 and P2) and stabilize the sprint code  Document known issues list prior to the sprint demo  Schedule and present sprint demo  Conduct sprint review and retrospection, document lessons learned  Perform solution build and release management activities to support development and testing  Manage scope change  Conduct quality reviews for software code, builds and project  Document P3 and P4 issues and update Product Owner about addition to the product backlog  Consolidation of Sprint demo feedback |
| **Customer activities** The activities to be performed by the Customer | Participate in backlog working sessions  Review, prioritize and approve product backlog  Review and approve / reject Sprint user stories  Review and approve solution architecture  Test user stories on Sprint on Sprint basis as per the Acceptance Criteria and provide feedback  Detail user stories for upcoming development sprints  Approve Sprint backlog for next development Sprint  Review functional & technical design documents  Review test results  Participate in defect triage process  Attend key meetings (e.g. Sprint planning, daily stand-up, demo reviews)  Review and accept release deliverables  Plan and execute incremental UAT  Provide test data for Release Stabilize (Integration) & UAT  Resolve issues, mitigate risks, resolve dependencies related with Shell organization and other third parties involved in this project (e.g. Transporean).  Participate in scope change management process  Participate in managing escalation  Secure timely decisions from Executive Management on change requests, issues, dependencies, risks to minimize impact on project schedule & cost for causes beyond control of the project team  Participate in quality reviews (software build, software code, project)  Plan and execute on organization change management  Plan and prepare Shell IT for application maintenance and support  Plan and execute on methodology content authoring  Collect Sprint demo feedback and provide to Microsoft for consolidation  Coordination and implementation of front-end changes.  Cross Product Owner communication and enabling decision making and prioritization of PBIs |
| **Entry Criteria**  Allows the Sprint to begin | Approved Sprint backlog for the current sprint.  Approved solution architecture for the current sprint  No P1 issues exist to start the Sprint (e.g. readiness of development & test environment, team access to environments) |
| **Exit Criteria**  Allows the Sprint to conclude before moving to the next Sprint / phase | When the Definition of done is accomplished (to be defined during Sprint 0) |

### Stabilization and Final UAT

The below table outlines the activities to be undertaken that need to occur during final UAT and handover. This would consist of 2 weeks of Stabilization and 1 week of UAT.

| Category | Description |
| --- | --- |
| **Microsoft activities** The activities to be performed by Microsoft | Provide issue fix support for UAT  Participate in UAT defect triage and verify issues fixes  Attend UAT Test Review Meetings  Allocation of resources to solve P1 and P2 issues identified in the Final UAT Phase  Conduct regression tests before releasing code for UAT deployment  Update Solution Backlog  Prepare and ready UAT test environment  Prepare for Handover to Shell development teams |
| **Customer activities** The activities to be performed by the Customer | Support issue fix during UAT  Participate in UAT defect triage and verify issues fixes  Attend UAT Test Review Meetings  Review, prioritize and approve solution backlog  Provide UAT test data  Plan and execute UAT test cases  Conduct go/no go for pilot phase deployment  Accept solution code for pilot phase  User setup and access  User, metadata and content data load  Complete organization change management  Prepare and ready support (Help Desk, production support) organization  Prepare and ready infrastructure and operations support (maintenance, upgrades)  Conduct regression tests before releasing code for UAT deployment  Update Solution Backlog |
| **Entry Criteria**  Allows the Sprint to begin | As per Definition of Ready for Release to UAT (to be defined during Sprint 0) |
| **Exit Criteria**  Allows the Sprint to conclude before moving to the next Sprint / phase | As per Definition of Done for UAT (to be defined during Sprint 0). |
| **Dependencies** | Final UAT can be carried out only if all dependent solution components are built by other stakeholders (e.g. Shell Market-Hub, TMS, etc.) |

### Work Products / Deliverables

Microsoft will provide the following work products / service deliverables.

| Name | Description | Phase | Acceptance Required | Acceptance Criteria |
| --- | --- | --- | --- | --- |
| Prioritized backlog | A prioritized backlog that which is agreed with the customer and fits with the contracted capacity | Sprint 0 | No | Prioritized backlog agreed between the Shell Product Owner and the Microsoft Lead Project Manager. |
| High Level Design Document | A word document which accurately reflects high level design, API contracts, database models and design decisions taken in Sprint-0.  Word based Microsoft documentation format will be used. | Sprint 0 | No | Document reflects High Level Solution Architecture to be agreed in Sprint-0 and accepted by the Shell Lead Architect(s) before development sprints start. |
| Sprint completion report | This report lists the in-scope items that have been completed during the sprint, any planned work that was not completed, and any project risks or problems. This report is produced as an output of each sprint. | At the end of each Development Sprint | No | Not applicable |

### Deliverable acceptance process

During the project, Microsoft will submit certain deliverables (listed in the Approach section as deliverables with “Acceptance required?” equal to “Yes”) for the customer’s review and approval.

**Within three (3) business days of the date of submittal**, Shell is required to:

* **Accept the deliverable** by signing, dating, and returning a service deliverable acceptance form, which can be sent by email, or by using (or partially using) the deliverable.

Or

* **Reject the deliverable** by notifying Microsoft in writing; Shell must include a complete and objective list of reasons for rejection, being as specific as possible to enable a resolution to be reached.
* Review and acceptance of the solution or custom source code is based on completion and acceptance of UAT as described in the Testing and defect remediation section.

Deliverables shall be deemed accepted unless the written rejection notification is received by Microsoft in the timeframe specified.

If a rejection notification is received, Microsoft will correct problems with a deliverable that are in scope for the project (and documented in this SOW), after which if the customer still rejects the deliverable, this must be done through the escalation process described in Section 2.4.4 (Escalation Path).

Problems that are outside the scope of this project, baselined after Sprint 0, and feedback provided after a deliverable has been accepted will be addressed as a change request, managed as described in the Change management process section.

### Deployment

Microsoft will build a CI/CD process which will enable performing deployments across all defined environments. However, deployment to any environment other than development, test and UAT is the responsibility of the Customer. Customer should initiate deployments for the other environments.

## Sprint Principles

### Sprint Principles

*Please note that the principles outlined below are subject to alignment and agreement during Sprint 0.*

* User Stories should be “ready for development” at least 1 Sprint before the start of the Sprint
* Any user story which is not “ready for development” 1 Sprint prior to the start of development activities will be postponed to the next sprint by when open issues and/or dependencies will have been fixed
* Shell will comply with a turn-around time not exceeding two (2) Business days for requests for clarifications on functionality and/or questions regarding bugs subject to the active sprint.
* User Stories selected for the next Sprint should be prioritized discretely using numbers starting from 1. Number 1 takes the highest priority. An alternative approach could be to stack rank backlog in Azure DevOps Services. It is expected that one of these approaches would be jointly agreed and established as a standard practice for the project.
* No two user stories can have the same Priority number or backlog stack rank
* As scope is understood to take priority following Sprint 0, Release schedule is to be defined for the Release considering the available capacity and dependencies
* Business tests for the previous sprint should be completed by Shell Product owners / Business Analysts (BAs) / SMEs prior to the start of the next sprint. This will allow for development teams to formally make a closure on the previous sprint and report on sprint performance. In addition, for defects arising from previous sprint, the time can be utilized to plan and prioritize defects for inclusion to successive sprints.
* At the start of each sprint, each development team would discuss and report on critical issues which could hinder Sprint performance. Such issues should be addressed by the development team or escalated via the agreed channels where appropriate.
* Microsoft and Shell project teams should conduct regular and periodic Sprint retrospective. Sprint metrics should be studied, and lessons learned documented.
* Shell reviewers need to compile, prioritize and approve the Product Backlog (Epics, Features and User Stories) in a timely manner to avoid impact on start of development sprints subject to ‘ready for review’ and ‘complete’ criteria to be defined in Sprint 0.
* To improve sprint development quality, Microsoft and Shell project teams should agree on the right level of documentation around defects. At the minimum, such documentation shall cover defect description, classification (Priority) and the resolution provided during defect closure. The agreed level of documentation should be reviewed regularly after each sprint cycle to make certain that the right level of detail is captured. A proposed review cycle may include, but not be limited to:
  + After the Shell Product owners / Business Analysts (BAs) / SMEs have conducted the user story testing
  + As part of sprint retrospective
  + During execution of Test sprints

## Timeline

Planned capacity is estimated based on a project of 16 weeks duration. This consists of Sprint 0 (4 weeks) followed by 3 development sprints (of 3 weeks each) and ending with a Stabilization phase of 2 weeks and a Final UAT phase of 1 week duration. The specific timeline will be finalized during Sprint 0 and will be updated as part of core project management activities.

For each subsequent pilot/production release, appropriate release stabilization and UAT duration will be added in the project schedule. This is outside the scope of the MVP development.

## Project governance

The governance structure and processes the team will adhere to for the project are described in the following sections.

### Backlog item acceptance

Backlog items (user stories or bugs) do not require formal sign-off or Customer acceptance when they are completed by the development teams. However, customer approval of PBIs is integral part of the VSTS workflow, i.e. the Product Owner(s) will have to indicate that the Product Backlog item (PBI) is “Approved” and “Done” (Definition of Done is met). Definition of Ready is a formal Microsoft approval step indicating readiness to commit user stories to development.

Any defects found in a finished backlog item will be added to the product backlog as a bug and prioritized by the Customer product owner with the other backlog items. A finished backlog item may also prompt the Customer product owner to add additional backlog items to enhance the software. Shell would have to be cognizant that re-prioritization of backlog may result in additional, unexpected, capacity to deliver the work than what is contracted. In such a case, change request would be required under the process defined in section “2.4.5 – Change management process”.

### Project communication

The following will be used to communicate during the project:

* **Communication plan**: this document will describe the frequency, audience, and content of communication with the team members and stakeholders. It will be developed by Microsoft and the Customer as part of project planning.
* **Status reports**: the Microsoft project manager will prepare and issue regular status reports to project stakeholders per the frequency defined in the communication plan.
* **Status meetings**: the Microsoft project manager will schedule regular status meetings to review the overall project status, the acceptance of deliverables, and open problems and risks.
* **Sprint completion report**: the Microsoft project manager will compile sprint completion reports following the completion of each development sprint for distribution to both Customer and Microsoft management.
* **Sprint review meetings**: meetings will be held to review the overall project status, the project schedule, and open issues that were noted in the reports. It is important that Shell is sufficiently represented during such '**Sprint review meetings**' to align on expectations and remove impediments and enable change management to facilitate an efficient handover at end of release.

### Risk and issue management

The following general procedure will be used to manage active project issues and risks during the project:

* **Identify**: identify and document project issues (current problems) and risks (potential events that impact the project).
* **Analyze and prioritise**: assess the impact and determine the highest priority risks and issues that will be managed actively.
* **Plan and schedule**: decide how to manage high-priority risks and assign responsibility for risk management and problem resolution.
* **Track and report**: monitor and report the status of risks and issues and communicate issue resolution.
* **Control:** review the effectiveness of the risk and issue management actions.

Active issues and risks will be monitored and reassessed on a weekly basis. The status and documentation of active issues / risks will be updated on a regular basis to reflect the progress on actions items and impact on planned resolution / mitigation date. They will also be discussed in detail as part of the Sprint review meetings and weekly project status report reviews meetings.

### Escalation Process

The Microsoft Project Manager will escalate Project issues, risks and change requests to the Shell Project Manager, as necessary. The planned escalation process for the review and approval and/or dispute resolution is documented below and consists of three levels.

Project Managers will engage team members as needed to resolve blockers, risks or issues as they arise. Expected roles to engage will fall into one of technical, functional or management related and should be able to work in an ad hoc capacity to provide clarity or insight to remediate any issues.

* **Level 1 Project Team(s)**: Project Team members are expected to resolve team issues without having the need to escalate to the Project Management Team (as defined above). Typically, these issues are more related to day to day activities of the project teams. In case the team is not able to resolve the issue, it should be escalated to the Project Management Team.
* **Level 2 Project Management**: Any escalation that cannot be addressed by the Project Team members should be addressed by the Project Management Layer. In addition, this team is also responsible to resolve escalations with impact to the overall project (e.g. cost, schedule, resources). This team will escalate to the project executive sponsors if they cannot resolve the escalation
* **Level 3 Executive Steering committee**: Escalations that cannot be resolved by the Project Management Team are escalated to the Shell & Microsoft Executive Sponsor for resolution. This team is the final decision maker for escalations. Representation of the Executive Steering committee to be agree before commencement of the project and will include the Shell Account Delivery Executive in the first instance.

### Change management process

During the project, either party may request modifications to the services described in this SOW. These changes only take effect when the proposed change is agreed upon by both parties.

#### What is a ‘Change’?

* Any deviation to documented ‘Assumption(s)’, ‘Constraint(s)’, and / or Dependencies’, either in this contract, or any subsequent deliverables / agreements throughout the duration of this contract
* Any ‘Update’ to and / or ‘Deletion’ of referred scope (Section 1.2) in this SoW or the baselined project scope after completion of Sprint 0.
* Any extension / reduction either to project / sprint duration, or onsite resource(s)
* Any change in location of resources, from offshore to onsite (Amsterdam, Netherlands)
* Any change in environment(s), deployment location(s) and / or skills / technology required from what have been documented in this contract
* Any delays in project due to non-fulfillment of tasks, dependencies and / or activities from your end
* Any change in duration / time-period provided for review and / or acceptance of various deliverables and / or milestones
* Any adherence to local regulatory / statutory policies, not explicitly covered as ‘In Scope’ under this contract and/or which impact has not been accessed as part of the effort estimates
* Any change in responsibilities / activities between you and us as currently described in this SoW and accompanied Work Order

The change management process steps are:

* **The change is documented**: all change requests will be documented by Microsoft in a Microsoft change request form and submitted to the Customer. The change request form includes:
  + A description of the change.
  + The estimated effect of implementing the change.
* **The change is submitted**: the change request form will be provided to the Customer.
* **The change is accepted or rejected**: the Customer has 3 (three) business days to confirm the following to Microsoft:
  + Acceptance—the Customer must sign and return change request form.
  + Rejection—if the Customer does not want to proceed with the change or does not provide an approval within 3 (three) business days, no changes will be performed.

During the project, either party can request, in writing, additions, deletions, or modifications to the services described in this SOW (“change”). Approved changes will be managed through amendments and could lead to additional costs and schedule impacts. We shall have no obligation to commence work in connection with any change until the details of the change are agreed upon in an amendment signed by the authorized signatories from both parties.

Within 3 (three) consecutive business days of receipt of the proposed amendment, you must either indicate acceptance of the proposed change by signing the amendment or advise us not to perform the change. If you advise us not to perform the change, we will proceed with the original agreed upon services only. In the absence of your acceptance or rejection within the previously noted time frame, we will not perform the proposed change.

### Executive steering committee

The executive steering committee provides overall senior management oversight and strategic direction for the project. The executive steering committee for the project will meet according to the frequency defined in the communication plan and will include the roles listed in the following table. The responsibilities for the committee include:

* Making decisions about project strategic direction.
* Serving as a final arbiter of project issues.
* Approving significant change requests.

| Role | Organization |
| --- | --- |
| Project sponsor | Customer |
| Delivery Executive | Microsoft |

### Guiding principles

The escalation path and the related process will be discussed and finalized at the project kickoff, but the following general guiding principles are expected to apply:

* Significant project issues and risks, as well as material change requests that cannot be resolved by the core project team, will be escalated to the executive steering committee as the final decision maker. The expectation is that the executive steering committee will take positive action to get the issues resolved in a timely manner, accept or implement recommended mitigations for identified risks, or make final decisions on the disposition of proposed change requests.
* If a major unresolved item requires escalation prior to a scheduled executive steering committee meeting, a special meeting will be scheduled, or the item will be escalated to the committee in writing.
* It is understood and agreed that if the executive steering committee does not act to resolve items that are presented to it in a timely fashion, project schedule and cost slippage may result, which may result in additional change requests.

## Project completion

The project will be considered complete when at least one of the following conditions has been met:

* All Microsoft deliverables that require acceptance have been delivered and accepted (or deemed accepted).
* The Work Order has been terminated.

# Project organization

## Project roles and responsibilities

The key project roles and the responsibilities are as follows.

### Customer

| Role | Responsibilities | Commitment |
| --- | --- | --- |
| Project sponsor | Provide the estimated project commitment: part time  Make key project decisions.  Serve as a point of escalation to support clearing project roadblocks. | Part-time |
| Product owner | Provide the estimated project commitment: full time (could be per feature team)  Manage and prioritize the product backlog.  Serve as the primary person responsible for user story scope decisions during sprint planning.  Define acceptance criteria for work items, especially user stories.  Actively participate in all sprint reviews.  Serve as the single point of contact for decisions about product backlog items and prioritization.   * Provide quality documentation in user stories by following the user story template mutually agreed with Microsoft so that any requirement gaps can be minimized | Full-time |
| Project manager | Provide the estimated project commitment: full time  Manage and coordinate the overall project and deliver it on schedule.  Take responsibility for Customer resource allocation, risk management, project priorities, and communication to executive management.  Coordinate decisions within 3 business days, or according to an otherwise agreed-upon timeline. | Full-time |
| Security Team / Architect – Application, Data | * Provide Shell IT security policy inputs * Provide inputs, review and approve security architecture * Administer Shell’s Active Directory (Authentication, Identity Team, Authorization) | Part-time |
| Subject matter experts and stakeholders | Provide the estimated project commitment: part time  Participate in the rapid envisioning workshop.  Can provide guidance to the Customer product owner. | Part-time |
| Business Analysts (BAs) | Self‐organized to deliver potentially deployable solution to product owner for testing. The BAs works together to:   * Author business Epics, Feature and User Stories in Azure DevOps Services and ensure they meet definition of ready * Agree on how to deliver the approved backlog Items in a Sprint (Sprint Plan) * Work to deliver on scope of Sprint working through the Sprint backlog using user story priority & estimate * Participate in daily stand up meetings * Share ownership and accountability for Sprint success * Participate in Sprint demos and retrospective meetings * Test developed User Stories on sprint on sprint basis * Collaborate with each other and key stakeholders to resolve issues and risks for the sprint | Part-time |
| UAT Lead | * Prepare, plan and execute on UAT * Provision test data (all environments except development) * Coordinate with Project development team for UAT defect support * Coordinate with Build and Release manager for solution deployment for UAT execution | Part-time |

### Microsoft

| Role | Responsibilities | Availability |
| --- | --- | --- |
| Account delivery executive | Responsible for your overall satisfaction with our services.  Serve as the single point of contact for billing problems, personnel matters, contract extensions, and Microsoft Services project status.  Responsible for engagement governance.  Facilitate communication.  Manage resource allocation and help form the development team.  Implement project management. | Part-time |
| Lead Project Manager | Manage and coordinate the overall project.  Responsible for resource allocation, risk management, project priorities, and communication to executive management.  Manage day-to-day activities of the project.  Coordinate the activities of the development team to deliver according to the project schedule.  Facilitate status reviews with the Customer.  Help make sure that the development teams are focused on the right goals, and help it meet the critical success factors of the project.  Ensuring delivery in line with the contract – if changes are necessary, executing contract modifications  Budget and Schedule Forecasting  Project is staffed appropriately and for success  The overall project schedule  Tracking and maintaining Proof of Delivery  Logging all significant Decisions Made by both the customer and Microsoft  Holding a periodic status meeting with the customer  Doing monthly Project Progress Report (PPR) documentation and delivery  Issue and Risk Management  Team member on-boarding and off-boarding  Clarity of Roles and Responsibilities and that all responsibilities are covered by members of the project team with no gaps  The overall communication with the customer  Managing project dependencies  Managing and approving time and expenses per policy  Creating and maintaining an executive escalation path and managing escalations | Part-time |
| Lead Architect | Provide technical oversight.  Lead the discovery and capture of solution requirements, features, and capabilities into the solution backlog.  Verify whether Microsoft-recommended practices are being followed.  Responsible for overall solution design.  Help provide activities and work products that are related to the engagement. | Part-time |
| Solution Architect | Generate Solution Architecture and Technical Designs as a means of communication between delivery and customer teams  Provide decision analysis and resolution for core architecture topics and technical decisions  Inform and consult with Microsoft and Shell teams as a means to gain insight and consensus on design decisions  Perform POC’s as needed in Sprints to confirm functionality hypothesis before they accepted and entered into the backlog. | Full-time |
| Development Lead (Offshore) | Work with the architect to determine the engagement requirements and perform design activities.  Lead design activities whenever necessary.  Help make sure that the development teams follow defined processes and standards.  Design and Code Review  Coordinating security reviews  Ensuring that each development team packages and deploys a quality solution to the extent possible with available capacity. Microsoft will be transparent to Shell through Azure DevOps tool on all the known defects and mutually agree on the resolution plan  Ensuring that each development team is following appropriate engineering processes (ALM)  Working with the delivery architect to identify technical dependencies and constraints (e.g. 3rd party packages) and resolve technical issues  Understanding the User Stories that are being proposed for the workstream  Working with the Product Manager(s) / SMEs to understand the acceptance criteria for each user story  Ensuring task breakdown for a User Story is correct and sufficiently granular to measure progress  Ensuring User Stories for their workstream, once begun, continue unimpeded until Done  Providing timely awareness if a committed story cannot be completed in the sprint  Ensuring User Stories meet Acceptance Criteria before declaring them Resolved or Done  Ensuring User Story, Bug, and Task status is correctly reflected in VSTS | Full-time |
| Delivery Architect (Offshore) | Ensuring the solution follows enterprise and industry standards  Validating the feasibility of features and capabilities  Facilitating scope elicitation with customer (ex: Epics / Features / Stories)  Managing the solution scope and critical trade-off decisions  Working with the Development Lead(s) to identify technical dependencies and constraints (e.g. 3rd party packages) and resolve technical issues  Document Solution Architecture  Document overall Data Model and any necessary documentation  Responsible for the Security of the solution, making sure the solution is designed for security, and the Security Strategy | Part-time |
| Offshore Project Manager / Scrum Master | * Responsible for managing and coordinating the delivery * Responsible for risk management, project priorities, and communication * Manages day-to-day activities of project * Coordinates the activities of the development teams to deliver deliverables according to the project schedule | Full-time |
| Development Team (Offshore) | Developers dedicated to build designed solution by applying Microsoft recommended practices | Full-time |
| Test Lead and Testers (Offshore) | * Test Lead - leads the test team activities to confirm that the solution is developed according to the agreed upon acceptance criteria for Epics, Features and User Stories within the constraints of scope assumptions * Testers are responsible for conducting the system & release testing and certifying the Sprint scope for UAT, Stabilize deployment * Performance Testers - Consultants responsible for performing performance testing including load testing as per documented scenarios to meet agreed nonfunctional requirements. | Full-time |
| Build and Release Consultant (Offshore) | Responsible for overall quality of Sprint demo and release deployable builds – free of P1/P2 defects, acceptable level of unit test coverage, successful execution of build regression test.  Create and manage the on-shore and off-shore build process  Certify builds before releasing for testing  Coordinate with Release Manager to plan and execute on code branching and merge strategy  Oversee the off-shore Release Team | Full-time |
| InfoSec Architect (Offshore) | Supports Shell with security requirements especially around SOC 2 Type II and GDPR  Participates in review and enhancement of Non-functional requirements and user stories in collaboration with Shell’s Security team  Performs validation on user [input](https://liquid.microsoft.com/Web/Object/Read/ms.security/Requirements/Microsoft.Security.SystemsADM.10027) and [output](https://liquid.microsoft.com/Web/Object/Read/ms.security/Requirements/Microsoft.Security.SystemsADM.10083).  Advocates and incorporates applicable [Microsoft internal Security Development Lifecycle (SDL) requirements](https://liquid.microsoft.com/Web/Views/View/49816) as part of SDLC  Advocates the need to comply with SDL process | Part-time |
| TQA Architect (Offshore) | Reviews technical quality of project as well as reviews key deliverables from each phase.  Manages quality gates and evaluates deliverables for Microsoft recommended practices.  This is an independent review performed by consultants who are not involved in the development of the application. | Part-time |

# Customer responsibilities and project assumptions

## Customer responsibilities

In addition to Customer activities defined in the Project Approach section, the Customer is also required to:

* Provide information.
  + This includes accurate, timely, i.e. within 3 (three) business days or as mutually agreed upon, and complete information.
* Provide access to people and resources.
  + This includes access to knowledgeable Customer personnel, including business user representatives, and access to funding if additional budget is needed to deliver project scope.
  + Acquire and install the cloud capacity that is needed to support the environments as defined in the scope section of this SOW.
  + Shell will ensure dedicated availability of Product Owners and Subject Matter Experts (SMEs) throughout the Project to perform activities as documented in the Customer Roles & Responsibilities table.
* Provide access to systems.
  + This includes access to all necessary Customer work locations, networks, systems, and applications (remote and onsite).
* Provide a work environment.
  + This consists of suitable work spaces, including desks, chairs, and Internet access for the Microsoft resources who would be working out of Shell premises at Netherlands.
* Manage non-Microsoft resources.
  + The Customer will assume responsibility for the management of all Customer personnel and vendors who are not managed by Microsoft.
* Manage Solution Components
  + In Section 1.2.2 for those solution components which are indicated in Yellow and Red, Shell should follow their exemption process to make these components available for this project. It is expected that these components are approved and communicated to Microsoft before the start of the project. Should there be any delays with the approval, the project start would have to be deferred.
* Manage external dependencies.
  + The Customer will facilitate any interactions with related projects or programs to manage external project dependencies.
  + Troubleshoot systems that are not being developed by Microsoft.
  + Confirm regulatory compliance.
  + Provide any required product training to Shell and client staff
* Other general responsibilities, as applicable.
  + Monitor network activity.
  + Provide application support.
  + Responsible for the financial costs associated with hardware purchasing, software licensing, or purchasing of Microsoft or third-party tools.
  + Bug fixing and troubleshooting problems that are related to applications or other third-party software, hardware products, or applications that are not explicitly mentioned as in scope.
  + Prepare documentation about processes, standards, policies, or existing guidelines.
  + Plan, design, customize, enhance, troubleshoot, or resolve problems that are related, but not limited, to supporting the infrastructure listed here:
    - Firewalls.
    - Storage area networks.
    - Networks.
  + Design, install, and configure the environment (other than development and system testing).
  + Organizational change management.

## Project assumptions

The project scope, services, fees, timeline, and our detailed solution are based on the information provided by the Customer to date. During the project, the information and assumptions in this SOW will be validated, and if a material difference is present, this could result in Microsoft initiating a change request to cover additional work or extend the project duration. In addition, the following assumptions have been made:

* Work day:
  + The standard work day for the Microsoft project team is between 9 AM and 6 PM, Monday through Friday, local time (IST) where the team members are working.
* Standard holidays:
  + Observance of consultants’ country-of-residence holidays is assumed and has been factored into the project timeline.
  + The vacation and holiday schedule for the development teams would be discussed with Shell so as to manage impact of regional holidays around sprint planning and execution.
* Working location:
  + The physical location when not working remotely will be agreed between the Shell and Microsoft project managers.
  + For the workshops during Sprint 0 based on Shell’s request the assumption is that these would occur in one of Shell’s offices based in London.
* Remote working:
  + The Microsoft project team may perform services remotely.
  + If the Microsoft project team is required to be present at the Customer location on a weekly basis, resources will typically be on site for three nights and four days, arriving on a Monday and leaving on a Thursday.
* Language:
  + The e-mail templates will be multi-lingual. By default, Microsoft will support only English. The customer has responsibility to prepare, provide and test email templates in different languages, however, these languages should be limited to left-to-right languages. .
  + All project communications and documentation will be in English. Local language support and translations will be provided by the Customer.
* Staffing of Microsoft project resources:
  + If necessary, Microsoft will make staffing changes. These may include, but are not limited to, the number of resources, individuals, and project roles.
  + Changes to critical Microsoft onsite and offshore resources would be consulted with Shell but the sole discretion remains with Microsoft
  + We have presumed that most of the design and implementation work will be performed by Microsoft Consulting Services (MCS). We have, however, assumed some level of involvement from your personnel as detailed in the Customer responsibilities. We have not accounted for any internal costs of that involvement.
  + It may take up to 8 weeks for Microsoft Services to staff the development teams from the time the contract is signed.
  + Development Team Ramp-up and ramp-down
    - It may take up to 8 weeks for Microsoft Services to add a new development team to the project from the time such an addition has been approved commercially.
    - It would necessitate that Shell gives Microsoft a notice of not less than 2 weeks before a development team can be removed from the project.
  + More sprints can be introduced to continue delivering more scope items in addition to what is defined in “ 1.2 – Areas in scope”. Any changes on the number of sprints to be executed by Microsoft:
    - Should be communicated in written by Shell at least 4 weeks in advance.
    - Unless otherwise is communicated in written by Shell, scope and the number of sprints will remain same.
* Informal knowledge transfer:
  + Customer staff members who work alongside Microsoft staff will be provided with information knowledge transfer throughout the project. No formal training materials will be developed or delivered as part of this informal knowledge transfer.
* Azure DevOps:
  + Development Teams will capture the solution backlog in Azure DevOps.
  + As the project backlog will be maintained in the Customer Azure DevOps environment, Customer will have the full transparency to the data.
* User story decomposition to follow INVEST principle (exact requirements to be aligned during Sprint 0)
  + Independent - Avoid dependency. Combine or split story when appropriate
  + Negotiable - Stories should have right amount of detail.
  + Valuable - Show value to customer / stakeholder. Not focus on technology, developer centric value
  + Estimable - Story contains sufficient detail to estimate the effort
  + Small - Broken down such that its specific enough to work upon and can be completed within a sprint cycle
  + Testable - Acceptance criteria should be apparent.
* Checklist for a User Story: The checklist is used to check if the user story can be handed over to development (Definition of Ready).
  + It adheres to the INVEST principle mentioned above
  + The Acceptance criteria are understood and can be translated to proper test cases
  + User Story has been estimated and assigned a USP estimate (User Story Point) by the Microsoft development team
* Non-functional related:
  + Performance metrics will be discussed and documented as part of the Solution Architecture document. Release specific business test scenarios are delivered by Shell and metrics are defined. These are incorporated in the Performance Strategy / Test document.
  + Requirements around Availability, Security, Monitorability, & Upgradeability shall be discussed during Sprint 0 and incorporated in the solution architecture document.
* Microsoft will provide guidance to Shell on what it considers a sufficient level of detail for a User Story. If the User Stories are not specific or detailed enough, Microsoft will ask questions and promptly revert back to Shell for clarifications, who should provide answers within the timelines agreed in this SoW in order to ensure readiness of the User Story. User stories that does not meet Definition of Ready will not be considered by Microsoft for development. Definition of Ready will be finalized during Sprint 0.
* Test related:
  + Shell will provide inputs into identifying functional test cases for sprint testing, in line with Acceptance Criteria.
  + Shell will provide test data during release stabilize, performance testing and UAT test execution with Microsoft support to create and load test data.
  + Shell is responsible for UAT test planning and execution.
  + Shell will progressively provide UAT test cases applicable to each sprint scope, so that the same can be verified during stabilization of each release.
  + Scope of test case automation is limited to automated developer unit testing for select code base, select regression testing, and automated build verification testing to be defined. Additional test automation will be discussed during development sprints, and if required, will be managed through Change Management process.
  + Shell will approve test cases one sprint in advance.
* Other assumptions:
  + The Customer will assign a team to collaborate on the project with the Microsoft team, including the provision of a full-time product owner to define and decompose user stories and a full-time IT project manager, to manage Shell architecture inputs, for required co-ordination with Market-Hub Transporean, business testing, IRM and any other required assurance, to schedule key milestone events, such as Sprint 0 sign off, and for payment.
  + Shell will advise Microsoft of the availability and location of key team members at least one month before planned start so that appropriate plans can be made for co-location for project kick off and for key milestone events.
  + The Customer will meet the necessary requirements to help make sure the solution design meets regulatory requirements.
  + Azure services and Azure-supported Microsoft technologies will be used to develop the solution.
  + The components to be developed by Microsoft will be cloud-hosted.
  + Microsoft will not modify any existing code base that was not produced by the Microsoft development teams unless there is a proper handover provided by Shell.
  + All project resources will have the appropriate level of security access needed to complete project-related efforts.
  + Holidays, vacation, and training time have not been factored into this SOW.
  + All work is to be contiguously scheduled. Any breaks in the engagement calendar must be scheduled four weeks in advance, or it will be billed without interruption.
  + Formal knowledge transfer will be required from Shell team to Microsoft team
  + No custom development or effort is required for work related to content display and/or content translation.
  + Microsoft is not responsible for testing the content of the application.
  + Pilot and post go-live support is currently not in scope of work for this Statement of Work
  + Service Bus communications will be secured using SAS tokens.
  + No localization support is required within the scope of the MVP. Should a need emerge, e.g. support for additional languages—it will be added to the product backlog for the Product Owner to prioritize via the Change Management process