**HAPPIEST MINDS TECHNOLOGIES LTD.**

**MSIL Smart Spares Digital Twin Platform Proposal**



S

**Copyright Information**

This document is the exclusive property of Happiest Minds Technologies Ltd. (“Happiest Minds”). The recipient agrees that they may not copy, transmit, use, or disclose the confidential and proprietary information in this document by any means without the expressed and written consent of Happiest Minds. By accepting a copy, the recipient agrees to adhere to these conditions to the confidentiality of Happiest Minds practices and procedures.

**Confidentiality Clause**

This document is being submitted to MSIL Enterprises Ltd by Happiest Minds Technologies Ltd. on the understanding that the contents of this document will not be divulged to any third party without the express written consent of the parties. It is also understood that the parties will not divulge any confidential information about MSIL that it may have access to during this interaction.

**Disclaimer**

This document has been prepared based on the information provided by MSILL. Wherever proposed, the solutions and/or services mentioned are based on the requirements defined and understood by us at the time of preparing this document. While every effort has been made to make this document as accurate as possible, there might be changes to the document based on the subsequent discussions.

[1 Executive Summary 3](#_Toc132310175)

[2 Requirement Details 6](#_Toc132310176)

[2.1 Actors / Users 7](#_Toc132310177)

[2.2 Functional requirement 8](#_Toc132310178)

[2.3 In Scope 10](#_Toc132310179)

[2.4 Out of Scope 11](#_Toc132310180)

[2.5 Supported OS and Browser versions 11](#_Toc132310181)

[2.6 Assumptions 11](#_Toc132310182)

[2.7 Dependencies on MSIL 12](#_Toc132310183)

[3 Solution Consideration 13](#_Toc132310184)

[3.1 Solution Architecture 13](#_Toc132310185)

[3.2 Data and analytics 25](#_Toc132310186)

[3.3 Non-functional Requirements 26](#_Toc132310187)

[3.4 Testing Approach 27](#_Toc132310188)

[3.5 Proposed DevOps Approach 30](#_Toc132310189)

[3.6 Technology Stack – Indicative 34](#_Toc132310190)

[4 Training 34](#_Toc132310191)

[5 Project Delivery Approach 35](#_Toc132310192)

[6 Governance Model 37](#_Toc132310193)

[6.1 Project Communication Model 37](#_Toc132310194)

[6.2 Execution Schedule and Deliverables 37](#_Toc132310195)

[6.3 Risk and Mitigation Plan 40](#_Toc132310196)

[6.4 Change Request Management 40](#_Toc132310197)

[6.5 Acceptance Criteria 41](#_Toc132310198)

[7 Proposed Team Structure 41](#_Toc132310199)

[7.1 Key Team Members (Indicative) 42](#_Toc132310200)

[8 Commercial 42](#_Toc132310201)

[9 Abbreviations: 44](#_Toc132310202)

# Executive Summary

Happiest Minds would like to thank MSIL for giving us this opportunity to bid for the RFP for their MSIL Smart Spares Platform. We understand MSIL intends to build a Smart Spares warehouse digital twin, to provide a platform to make parts and accessories available to dealers and distributors network.

The smart spares warehouse application shall display all relevant data points pertaining to the spares warehouse operations in real-time, to enable enhanced data visibility that helps in making faster and more data driven decisions.

The digital twin application at a high level is a replica of the actual warehouse process displaying data points encompassing Receipt, Binning, Picking, Inventory and Dispatch operations

**Our approach for MSIL**

We appreciate the information shared through the RFQ document and the responses to our queries. Happiest Minds, through our proven experience of building and managing end to end platforms, are proposing a comprehensive solution for MSIL.

* Robust platform providing end to end solution in Spares warehouse management.
* Understanding of the end-to-end ecosystem, existing inputs like reports and data sources
* Phased approach for the program where we will work closely with MSIL.
  + Discovery phase: To finalize the requirements and understand approach for individual components like dock management, storage capacity management, inventory management, binning etc
  + MVP-based implementation approach with clear outcomes – across 3 MVP phases by XX years – to achieve MSIL goals.
  + User-experience-based approach
  + Support in Pilot-based rollout and complete rollout
* Experienced team who has built and managed robust platforms with mix of technology specialists (AWS, Data integration, API management, DevOps) and domain consultants in the auto spare parts space, program management along with MSIL team – we propose a squad-based model to be engaged with MSIL through the program.
* We are an existing partner to MSIL and bring in awareness of the processes, standards, methodologies – including systems and areas of integration.
* Ready team excited to initiate the project

​ **Our Experience**

Happiest Minds brings proven experience across Industrial & Manufacturing customers across the value chain and with customers in electrification space. Our experience ranges from building robust platforms (IOT, Data Platforms, Connected Vehicle platforms, Experience Platforms, eCommerce, ERP platforms), connected thousands of devices/ vehicles and building robust integrations with the customer systems/ ecosystem partners.

Details of some of the key case studies are provided below:

1. **Platform Setup and Predictive Asset Maintenance for a large US-based power management company**

The client’s flagship product helps industrial users of UPS systems mitigate the risk of power component failure. With the possibility of downtime costs exceeding $1 million per hour, there was a need to transform in-service power component monitoring from a reactive to proactive model.

Happiest Minds helped the client build an IoT platform that stores, monitors, and analyzes critical real time data consisting of approximately 200 measurable parameters from subscribing devices. Telemetry data consisting of 50 different metrics is used for real-time monitoring and predictive maintenance.

With the new approach, the client can now drill down on individual UPS systems for predictive & prescriptive maintenance, provide real-time notifications to customers and support both brownfield and greenfield devices. The client’s application costs are estimated to be reduced by over 60% and incremental sales growth is projected to be 200% in the mature year.

1. **Digital Partner for a leading high-growth electrified powertrain company, providing hybrid and electric technology for Class 8 trucks.**

The client was looking at digital initiatives across the value chain.

1. The client had a lot of data coming in from vehicle telemetry and other sources, without clear data management.
2. Quality inspections across various stages are being managed in excel or paper-based methods and the client was looking to have a complete workflow-driven approach.

Happiest Minds started with assessment of their existing Data Platform and provide recommendations.

We are currently implementing automation & integration into Data Lake (key areas: Truck telemetry data integration, Master Truck data management, finance report automation)

We engaged with the client to understand the current process and evaluate the fitment into a COTS product (NetSuite) and provide guidance on OOTB features and identify the customizations required. The process spanned across quality inspection on work order completion, PO receipt and customer Return Merchandise Authorization (RMA) return receipt.  We are currently engaged with the client for the implementation phase.

1. **ERP Transformation for Leading electric vehicle’s hardware/software Manufacturer**

The client was looking for a reliable, scalable solution for their ERP platform. Happiest Minds implemented end to end capabilities in ERP - Order to Cash, Procure to Pay, Basic

Manufacturing, Advance Inventory. This provides:

* Greater visibility and control over Sales process & Finance reporting
* Billing and tracking transparency of sales process
* Consolidation of international subsidiary locations and facilitated multi-currency transactions.

1. **Targeted advertising & stock out alerts at fuel stations for leading Industrial products and applications provider in retail fueling & Smart Spares station space.**

The client was looking for an integrated solution to provide seamless consumer journey between the forecourt and instore at a fueling station.

Happiest Minds partnered with the client to enable customer engagement at the gas/ Smart Spares stations. A customer gets identified at the fuel station and is presented with Products recommendations enriched by Customer Insights and Advertisements (offers at store) at the Dispenser display, kiosk display or Shelve display from there he can go ahead and add Products to his cart and eventually buy it. ​

The system should support product stockout identification and notification along with customer checkout notification for the instore clerks. ​The system should have APIs that allow site managers to setup/configure a fueling site, control, and authorize fueling transactions. ​

This platform improved sales by providing a mechanism for displaying targeted ads on dispenser/ kiosk​ and provided better Inventory management.

*Happiest Minds is excited to respond for this RFP from MSIL and we believe we are well placed to partner with MSIL on this journey. We look forward to engaging with you on the program.*

# Requirement Details

The MSIL Smart Spares platform will be developed to cater to the needs of MSIL after sales spare part operations in the areas managing receipt, inventory, binning, and dispatch.

Below mentioned features will be built into the digital twin platform

* Receipt: Involving data points pertaining to receipt of material from different vendors, dock in and dock out of vehicles carrying the material.
* Binning: Involves data pertaining to inward stock at respective locations. This comprises of track part order status, oldest pending date status and binned carton status.
* Picking: Picking involves picking the required material from respective locations. This data view shows a list of picking locations and to track VOR (vehicle off road) orders
* Inventory: Provide a real time stock visibility to serve order more efficiently. Dashboard view of slow- and fast-moving parts.
* Dispatch: Assignment of consignees for dispatch after picking and consolidation. Show service trend analysis, vehicle loading status, docking management etc.

## *Actors / Users*

The below cross functional diagram summarizes clarity in roles and responsibilities of users to play during various stages.

|  |  |
| --- | --- |
| Actor | Description |
| Warehouse Admin | Super user |
| Floor Coordinator | View only |
| Consignee |  |
|  |  |

## Functional requirement

This section details MSIL’s Functional requirements with use cases, including features categorization as per recommended phases.

|  |  |
| --- | --- |
| **Module** | **Features** |
| Receipt | **Gate In-> (CL03)**   1. Security team at the gate verifies the parts line send by the vendor by refereeing to the DI and Vendor Invoice. 2. Once the verification is done then the truck process for GATE IN has been completed and It moves to next step. 3. Each truck has some TAT to perform if not then color coded indication should be reflected over the screen such as:  * On time : Green * Slightly More time to threshold: Yellow * Delay : Red   **Dock Management for receipt (manual upload in excel format)**   1. **Add Vehicle/truck** 2. for the first time in a day user have to **upload the Load Vehicle Plan** 3. It shall contain : Dock number/Plan Number/TPT Type/TPT Vehicle No.   **Update Vehicle Plan**   1. Once the plan for the day has been uploaded the user shall have the option to update it.   **Dock In-> (Manual upload in excel format)**   1. It involves the vehicle details to be uploaded through the DOCK MANAGEMENT. 2. Once the Truck reaches to the docking station (Warehouse) then the unloading team starts unloading the Part. 3. Docking involves the unloading of material at warehouse.   **Dock Out->**   1. This process signifies that the required parts from the particular vendor has been successfully unloaded from the truck. 2. The warehouse team ensures that all the material as per the vendor invoice and MSIL DI has been received. 3. If the warehouse team finds any deviation in the received parts which is not as per their DI then warehouse team maintains the record for the same.   **Gate Out-> (CL03)**   1. And once the truck is empty then it shall start moving towards the gate for GATE OUT process. 2. The user shall be provided various report which needs to be reflected on the screen:  * Dock Management * Rejection Management * Part Order Status * TAT Monitoring * Truck Tracking   3. The data for the reports should be ingested on the Smart Spear Warehouse portal and need to be represented as per the given table in RFP or can be later decided. |
| Binning | This process involves keeping of inward stock (received from the vendor) at respective location.   1. The warehouse teams are responsible for binning the received part line cartons.   **Rewards & recognition**   1. Any binner who has placed highest number of cartons in a specified duration then that binner shall be eligible or reward & Recognition. 2. The name of the binner employee shall be reflected. 3. To track Binning process there shall be various statistics & report which reflect the binning status:  * Part Order Status * Oldest Pending Status * Binning * Binned Carton * Record & Recognition |
| Picking | This process takes place when the parts need to be dispatched to any Consignee against their order.   1. **Assigning of Picking Ticket** (clickable action need to accommodate on the smart warehouse portal) 2. Picking is handled by Line worker who pick the parts to be loaded on to the truck based on the TICKETS they have been assigned. 3. User shall have the option to select the assign ticket to the Line worker along with the priority of the status. 4. User shall have the option to select the line worker for picking. 5. **Ticket Color Coding** (yet to be decided by MSIL Team) 6. Ticket are being assigned based on the priority, which can have below mentioned proposed colors:  * Red : High Priority * Yellow : Moderate * Green : Relaxed   7. **Reward & recognition**  The highest picker has t be selected by the respective supervisor and the name of that employee shall reflect on the portal screen.  Picking involves various report which can explain about the in process and picked orders such as:   * Picking * Pick Ticket Status * VOR/SOS/MOR Order status   There shall be an **indicative color coding (proposed)**, which reflects the status of the picked part against any type of order from respective dealer.   * **Red** : Needs focus and highest concerns. CALL icon shall indicate if it cross the threshold TAT * **Orange** :This also indicate that part/order picking status has slightly crossed threshold TAT * **Yellow** : Its show that the pick status has reached the threshold TAT * **Green** : It indicate that picking process has some time in hand and is well within TAT |
| Inventory | A system based storage Capacity Dashboard need to be provided which fetches the metadata from SKU\_Master, CMT & Inventory.   1. There shall be a floor layout display for the user to see the real time status of the stock storage location for respective floor. 2. This floor layout shall be robust and if required in future can be modified or can be change as per the MSIL requirement. 3. Generally, it shall also be referred to see the Fast Moving & Slow Moving stock status and if required can be placed accordingly. 4. Rearranging/replacing the fast Moving stocking nearer to the dock when help reduce picking & loading time for line workers.   There can be various report that are being user to track inventory status:   * **Inventory Monitoring**: to indicate real time stock level status n the respective floor. * **Part Running Out of Stock:** It shall have the status of the ordering of the parts, stock for which is in critical level. * **Same Part at Multiple Location:** Visual representation of same part being at various locations.   (user shall have the option to have a filter by part)   * **Floor-wise FMS analysis :** to track Fast-Moving & Slow-Moving stock, which may help user to take decision for replacement of parts accordingly. |
| Dispatch | Dispatch comes in picture when the picking is done then material must be loaded to the truck for the dispatch to respective consignee.   1. A similar kind of data tracking such as receipt (inward) shall be there for dispatch (outward) which involves TAT tracking of:   **MSIL Gate In -> Dock In-> Sequence->Dock Out -> MSIL Gate Out**   1. **Dock Management (manual upload in excel format)**   **Add Vehicle/truck**  for the first time in a day user have to **upload the Load Vehicle Plan**  It shall contain: Dock number/Plan Number/TPT Type/TPT Vehicle No.   1. **Update Vehicle Plan**   Once the plan for the day has been uploaded the user shall have the option to update it.      User shall have the option to see the status through various filters:   * **Consignee** : Who’s parts need to be dispatched * **Warehouse** : How many truck are at warehouse * **Transporter** : trucks related to transporter * **Invoice/Gate Pass** : how many has been invoices and given gate pass * **Inside MSIL :** Use shall be able to see the truck if it has entered the MSIL gate * **At Dock**  **:** Indicate the truck location at docking point * **Loading : indicate** the truck at loading station * **Outside MSIL :**indicates the gate-out trucks * **TAT :**User shall be able to see TAT for each Truck**s** * **ALL** :All truck at once * **Regular/NCR/Courier:** Consignee order is for regular/NCR/courier   The user shall also be provided with various reports:   * Service Ration Trend Analysis * TAT Monitoring * Vehicle Loading Status * Dock Management * Truck Tracking |
| Sequence | **Update Sequence number:**   1. User shall be able to update the sequence number as per the loading plan for the vehicle being loaded for particular Consignee against their order.   **Check Status:**   1. The user shall be able to check the sequence number that has been performed and which sequence number is in process. 2. Each sequence number shall be linked with the TAT and should reflect the TAT status for particular sequence accordingly, such as:  * On time : Green * Slightly More time to threshold: Yellow * Delay : Red |
| Order Picking Status | 1. If the order picking status crosses the threshold and reaches way beyond the set TAT then it needs to have a Call to Action. 2. Delays in picking status : shall be integrated with an automatic SMS/E Mail trigger to the floor coordinator |

**Integrations**

Below are the integrations which are considered for MVP 1.0.

1. MSIL Data source for spare namely CL03, Space, Smart, WMS
2. Email/ sms notifications
3. MSIL reporting service

## In Scope

* Design and Development of smart spares digital twin system
* UI/UX design
* Functional Testing
* API Test Automation
* Application Security (OWSAP 10)
* Performance Testing (10 key services)
* Setup the Infrastructure on AWS cloud (3 environments) along with MSIL DevOps team
* Support for English Language only

## Out of Scope

* Build a Digital twin platform with 3D models-based visualization
* Building Data Lake from Source dataset
* UI Test Automation
* PEN Testing (separate engagement)
* User Manual, SOP, Runbook
* Post-Production Application and Infrastructure Support (separate engagement)
* Any tool/ third party integration other than mentioned on the requirement section.
* Any items not mentioned in the “In Scope” section.

## Supported OS and Browser versions

The user interface will be supported and tested on the “latest Major.Minor version and Major.Minor– 1” at the start of the project for the following browsers:

|  |  |
| --- | --- |
| **Browser** | **Windows Desktop OS version** |
| Google’s Chrome | Windows 10 |
| Firefox | Windows 10 |

The below mentioned screen resolution will not have horizontal scroll bar:

* + Resolution equal to or above 1280 x 768 pixels (Desktop/Laptop only)

## Assumptions

* + MSIL will provide the CL03 / WMS / SMART / SPACE required data source and access details to fetch data on regular interval.

## Dependencies on MSIL

|  |  |
| --- | --- |
| **Dependency Factors** | **To be available by** |
| Product owner to discuss the requirement, user flows, features of smart spare service, clarification, and review of the features and user stories | Start of the project |
| Technical SPOC for feedback and clarifications | Start of the project |
| API documentation and data source access details for Staging Env for CL03,WMS, SMART, SPACE systems | Discovery Phase |
| API documentation, SDK and Staging Env for MSIL Identity Management | As needed during the project |
| Code repository access | Start of the project |
| MSIL to provide AWS accounts and environment | Start of the project |
| Provide any coding guideline or standards that happiest minds must follow for component implementation | Start of the project |
| Test strategy /plan/ cases shall be reviewed and accepted by MSIL before test execution starts. | As needed during the project |
| Review UAT Test Case document | As needed during the project |
| Response to Happiest Minds’ queries within two business days. Any delays in response might have an impact on the effort and schedule & will be treated as a Change Request, | As needed during the project |
| Feedback on the deliverables for each milestone must be provided within 3 days of the delivery by Happiest Minds, else, they shall be deemed accepted | During entire project engagement |
|  |  |

# Solution Consideration

## Solution Architecture



The above architecture diagram depicts high level architecture of the solution. The application will be deployed in AWS Cloud. The following are the major components of the system.

* + 1. Microservices / REST APIs

The application will have a set of well-defined microservices. The reason for using microservices is mainly to have a decoupled architecture that will help to achieve scalability, modularity, and maintainability. The application will have well-defined REST APIs that can be consumed by the presentation layer. These REST APIs can be easily integrated with other MSIL or 3rd party systems. All the services will be deployed in Kubernetes cluster.

* + 1. Smart Spare Service
* Cloud First Design
* Microservices / API First
* Scalable
* Modularity
* Maintainability
* Single Page Application for web app implementation
* Securable
* Performance
* Use existing MSIL Authentication System

**Platform Services**

Application will have a set of well-defined microservices which will be deployed on Serverless Architecture

* All the REST APIs will be exposed through API gateway.
* All API responses will be in JSON format except report CSV download.
* APIs will be secured through access token (JWT) which will be passed over the auth header on request. All communications are over HTTPS.
* Microservice reference archetype standardizes the services.
* A combination of both event-driven and REST approaches will be considered for interservice communication and third-party integration.
* All frequently accessed data will be stored in cache for improving API latency.
* All microservices logs will be pushed to centralized logging system.
  + 1. Data Set Integration ETL Job Service

This service will have well-defined AWS Glue jobs which will be invoked regular interval pull latest data set from Data lake which built based on CL03 / WMS / SMART / SPACE data sources for Part Order Status, TAT Monitoring, DI & Rejection Status, Oldest Binning date Status, Binned Carton Status, Binning Rewards, Picking Performance, VOR Status, Inventory level monitoring, Floor wise FMS analysis, Same part in Multiple location, MGP Report, Loading Plan, Gate In and Gate Out Report, WMS Sequence Data Tracking.

* + 1. AWS API Gateway

All the traffic (Mobile, CSMS web interface) to access the CSMS platform passes through the API gateway. Authenticating with MSIL AD to verify the MSIL employee will be integrated with API gateway.

* + 1. Authentication &Authorization

1. Authentication for MSIL users to access spare sparts platform will be done using MSIL Active Directory (AD).
2. Once the authentication is successful, a JWT token will be generated which will be used for all subsequent API calls.
3. RBAC service will be used for providing role-based access to the system based on permissions.
   * 1. Data set integration Service

This service will be developed as microservice which interact with MSIL data lake platform which built based on CL03 / WMS / SMART / SPACE data sources for Part Order Status, TAT Monitoring, DI & Rejection Status, Oldest Binning date Status, Binned Carton Status, Binning Rewards, Picking Performance, VOR Status, Inventory level monitoring, Floor wise FMS analysis, Same part in Multiple location, MGP Report, Loading Plan, Gate In and Gate Out Report, WMS Sequence Data Tracking

* + 1. Smart Spares Web Application

Smart Spares Web Application: Smart Spares presentation layer will be deployed in AWS S3 bucket along with CloudFront. Web application and backend server will communicate over HTTPS using REST APIs for exchanging data. The proposed solution will have the following major components in the presentation layer.

* User management & RBAC
* Receipt
* Binning
* Inventory
* Notification
* Reports
  + 1. User Management and RBAC service

The Admin, CSR and other MSIL users will be able to interact with Smart Spare through the presentation layer.

* There will be pre-defined super admin user in the system. The super admin will be able to create the users who can access the Smart Spares.
* Application will have a provision to create the roles and attach permissions to the roles.
* Access to the resources will be based on the role of the user.
* All permissions to the applications will be directly uploaded onto the database.
* Application will have the provision to allocate the roles for each user. Each user will have the role assigned in the system.
* System checks the role of the user and gives the permissions to access the resources based on the role.
* Middleware will validate the role of the user and check the permission for the role whether the user can access the resources.
* The presentation layer will be different fordifferent sets of users based on the role.
* This service will also have the provision to reset the password for the user.
* There will be a password expiry policy in place. Users will have to change the password before the expiry. Users can also change the password at any point in time. This will be integrated with MSIL.
* There will be a provision to reset the password if user forgets it. The user can enter their email-id, the system checks the validity of the user, this will be achieved through the MSIL system.
  + 1. Notification Service

The responsibility of the mailer service is to send the email to the person concerned. System will send the email in the following cases.

* Assign consignee notification for dispatch
* Send email to floor coordinator for pick up.
* Send SMS

In the above scenarios, the system will trigger the email request which will be pushed to AWS SQS. The mailer service will consume the SQS queue and send email through the AWS SES service. There will be a pre-defined email template for the above scenarios.

## Non-functional Requirements

|  |  |
| --- | --- |
| **Availability** | Application will be hosted on the serverless container, This will provide continuous and uninterrupted service, making the system highly available. |
| **Reliability** | Serverless computing offers a number of advantages over traditional cloud-based or server-centric infrastructure. For many developers, serverless architectures offer greater scalability, more flexibility, and quicker time to release, all at a reduced cost |
| **Scalability** | Serverless computing offers a number of advantages over traditional cloud-based or server-centric infrastructure. For many developers, serverless architectures offer greater scalability, more flexibility, and quicker time to release, all at a reduced cost |
| **Performance** | Based on the load on the service, serverless container will serve the requests among the services. |
| **Security** | Authentication and authorization mechanisms will be in place to access the system as well as the resources of the system. APIs will be secured through access token (JWT) which will be passed over the auth header on request and all communications are over HTTPS. All the sensitive information will be encrypted and stored in the database. |
| **Extensibility** | The proposed system will be a microservice architecture which will help us to add new features without affecting the existing services and enables use of any programming language and database which are suitable for that feature. |
| **Usability** | Smart Spares Warehouse Digital will be Single Page Application (SPA), this gives greater user experience and provides more dynamic interaction with presentation layer. Proper validation and error messages will be in place to guide the user. Design patterns will be considered while designing and coding the application. Caching mechanism will be used for a smooth transition between screens. |
| **Disaster Recovery** | Provided the information in DevOps section below |

## Testing Approach

* + 1. Testing Life Cycle

Requirement gathering

Testcase Management

Testcase Review

Testcase Feedback Incorporate

Feature Testing

Sanity Testing

Regression Testing

UAT

Support

* **Requirement gathering**: Feature-based information is gathered and analyzed. Based on the Analysis, the test plan is created.
* **Testcase Management**: Based on the analysis, the testcase will be drafted by for the OCPP test, OCPI test API, mobile app, and integration (API and Mobile) separately. Testcases will be reviewed internally, and feedback will be incorporated in the testcase.
* **Testing**: Testing of the OCPP protocol, OCPI protocol usage will be done using the simulator. For the admin portal-based Rest API’s, the testing will be carried out using postman. For the mobile app testing will be carried out using simulator/ real device. Integration testing (API and Mobile) will be carried out and together considered as feature testing. The bugs logged will be triaged on every sprint. For the bug fixes given, sanity testing will be performed. Before the UAT, regression testing of the features will be done.
* **UAT (User Acceptance Testing):** The testcase which are finalized by client (before 2 weeks of UAT). Features developed will be shared to client for the UAT, with no blocker, critical bugs. UAT Support will be given during UAT phase.
  + 1. Types of testing

|  |  |
| --- | --- |
| **Test Methodology** | **Description** |
| Protocol Testing | Validation of OCPP and OCPI is done using simulators |
| Installation tests (Mobile) | Validation of the application installation, upgradation & uninstallation would be performed on all the devices mentioned. |
| Rest API Testing (API) | APIs developed by happiest minds will be tested in terms of positive, negative, business logic based for admin portal. |
| User Interface testing (Mobile App + CSMS) | Testing of menu options, buttons and navigation flow of the application would be validated. |
| We would also be validating the look and feel as per the wireframe and existing application screens. |
| Integration Testing (API + Mobile) | Integration of the protocol, API’s with the mobile application will be tested. |
| User Scenario/ Business Logic  (API + Mobile) | Testing the application as per business logic and with focus on end user perspective would be considered |
| Compatibility testing (Mobile) | We would be executing test cases on all the devices and OS versions as mentioned |
| Interruption testing (Mobile) | Testing Phone call, notification, device alert, device shutdown when the application is running. This would be performed on latest version on Android (V14.x) and iOS (V16.x) |
| Field Testing Support | The field testing of OCPP & OCPI protocol and CSMS platform will be performed by MSIL for the developed feature using actual Charging station/points and other CPOs. Happiest minds will provide support to MSIL during these testing |
| Application Security Testing | Web Application Security Testing shall be planned for OWASP Top 10 Vulnerabilities |
| Performance Testing | Rest API’s will undergo performance testing |

* + 1. Test level Analysis

Testing will be done in the following levels.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Criteria** | **Dev Environment** | **QA Environment** | **Bug Fixes** | **Before UAT** | **Pre- Production** | **Production** |
| Sprint Level | (Story Level) | (Story Level) |  |  |  |  |
| Rest API's |  | Regression | Sanity | Regression |  |  |
| Web portal |  | Regression | Sanity | Regression | Sanity | Sanity |
| Mobile application |  | Regression | Sanity | Regression | Sanity | Sanity |
| Integration Testing |  | Regression | Sanity | Regression |  |  |
| Performance Testing |  |  |  | Yes |  |  |
| Security Testing |  |  |  | Yes |  |  |
| Field Testing |  |  |  |  | Sanity |  |

* + 1. Performance testing

The performance testing shall be carried out as per the identified performance requirements between Happiest Minds and MSIL

#### 3.3.4.1 Objective

The main objective of Performance testing is as following (not limited to)

* Establish a benchmark for each of the user flow and for the system (in terms of Req/sec, CPU load, concurrent users,….)
* To identify and fix any performance bottlenecks in the system.
* To identify the scalability patterns of the system.

#### 3.3.4.2 Test approach

* HappiestMinds performance testing team will start with collection of requirements and identification of user workflows and individual APIs.
* HappiestMinds team will setup a performance testing environment similar to production.
* Perform a Tool fitment exercise to identify the right set of tool for performance testing and monitoring.
* Following tests will be performed to achieve the above said performance objectives,

##### 3.3.4.2.1 Atomic Test

An atomic test will be executed at the start of the assignment before any other test is executed. This test will load a single user on the identified critical scenario. The single user will execute the critical scenarios 3 times (3 iterations). The average of the 3 resultant response times will be reported in the atomic test results. A brief report detailing the observed response times of all the transactions involved in the critical scenario will be provided at the end of atomic testing. The objective of atomic testing is to verify the performance of the application under test when subjected to a single user load for a short period of time. The atomic test results can also be used as the first cut for baseline result comparison against the expected response times. This will be the system’s best baseline, as it is executed with zero concurrency. Successful completion of the atomic test will be a pre-requisite for the execution of load tests.

##### 3.3.4.2.2 Load & Scalability Test

Load testing of the application will be done once post the execution of the atomic test. The tests will be conducted by simulation of a pre-determined virtual user load on the servers. The load test will be executed with the identified user load in various sets (like, 50 users, 100 users, 200 users…). This will give results for each level of expected load. The load test will generally be run for a period of 1 hour with 5-10 minutes of ramp up time and 50 minutes of steady state. After the run of each test, the results will be captured, collated and reported. These reports can be used as the baseline set of performance results for performance tests that maybe executed in the future. These results can also be used for comparison with the atomic test results that will provide the deviation in performance of the system on a “‘multiple users’ versus ‘single user’” mode.

#### 3.3.4.3 Testing Cycles

* There will be an overall three cycles of performance testing to benchmark the performance of the system and to validate the performance fixes.
* Apart from that, on need basis to validate any performance fixes additional individual testing can be carried out, but there will be impact on the schedule and effort.

#### 3.3.4.4 Tools

* A Detailed POC-Tool fitment exercise will be carried out to finalize the tool.

#### 3.3.4.5 Reporting Metrics & Performance Expectations

* Generally, all the system metrics (like CPU, Memory, Network, Disk…) and all the request level metrics (response time, request/sec, errors, exceptions) will be monitored and reported.
* Apart from that, any other identified metrics during requirement stage will also be captured and this section will be updated later accordingly.

#### 3.3.4.6 Data Setup

* Ideally a production-like database has to be used for performance testing and hence testing will simulate necessary data to simulate the real-world load in the system. as we get more clarity on the requirements and system details, we will update this section in detail.

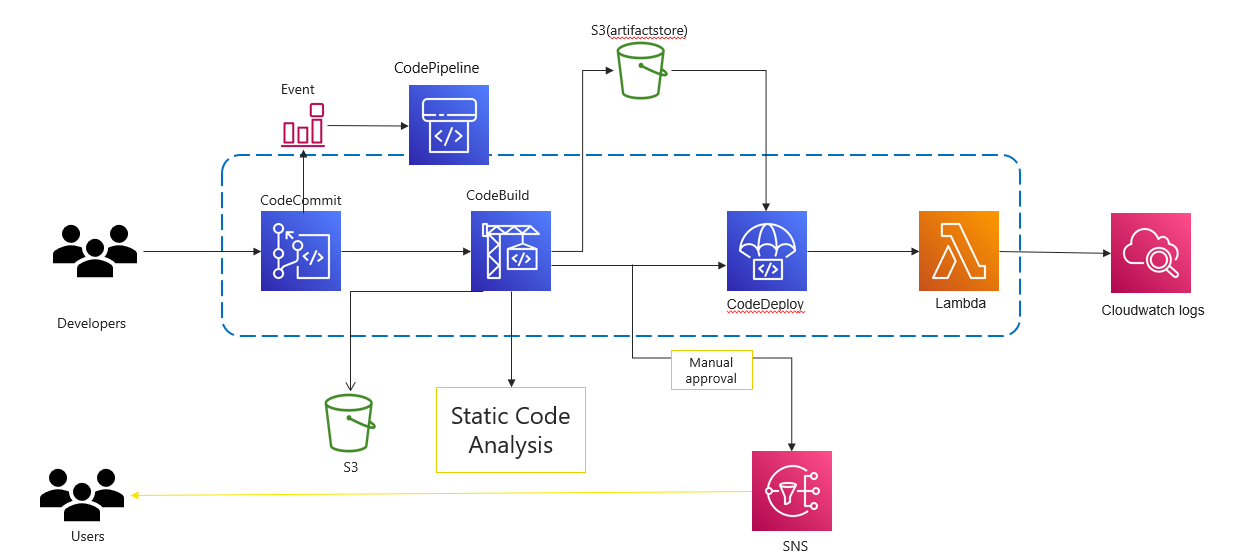
Few of the above-mentioned points are generic and shall be updated / concluded based on the benchmarking done between Happiest Minds and MSIL teams.

* + 1. Performance testing
* Web Application Security Testing shall be planned for OWASP Top 10 Vulnerabilities
* Automated Security Scans to identify existing vulnerabilities & Manual security testing of Web Application shall be conducted.
* No Infrastructure or Penetration testing is considered in the current scope of the project as the infrastructure management is not considered in the scope of this proposal.
* Remediation Support and to fix the identified vulnerabilities to the development team.

## Proposed DevOps Approach

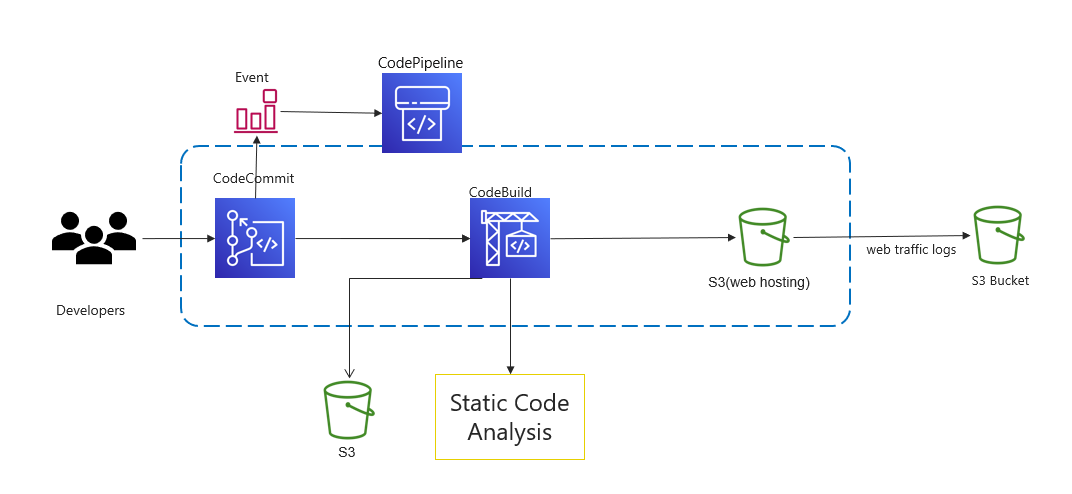
* + 1. CI/CD Approach for Lambda

Proposed approach for the Nodejs lambda deployment.



The main steps are as follows:

* When a user commits the code to a CodeCommit repository, a CloudWatch event is generated which triggers CodePipeline.
* CodeBuild scans the code with an SAST tool, and the test results would be uploaded to the AWS S3 bucket.
* CodeBuild packages the build and uploads the artifacts to an AWS S3 bucket.
* The approval stage is triggered, once the build is done and an email is sent to the approver for action. This is fully optional.
* CodeDeploy is a deployment service that automates application deployments to Amazon serverless Lambda functions, approval CodeDeploy deploys the artifact to lambda.
* The lambda logs would be available on CloudWatch, monitor and analyze logs from an Amazon CloudWatch Logs log stream.
  + 1. CI/CD approach for ReactJS



The above-mentioned approach will be followed for the deployment of React JS application onto the S3 bucket:

* When a user commits the code to a CodeCommit repository, a CloudWatch event is generated which triggers CodePipeline.
* CodeBuild scans the code with an SAST tool, and the test results would be uploaded to the AWS S3 bucket.
* CodeBuild packages the build and uploads the artifacts to an S3 bucket, S3 buckets for static web hosting because they can serve objects over HTTP.
* The web traffic logs would be logged to the AWS S3 bucket which is created for logging alone.
  + 1. Infrastructure Automation

Infrastructure as Code (IaC) is an approach that enables organizations to automate their infrastructure deployment and management using code. **AWS CloudFormation** is a powerful IaC tool that allows organizations to model and provision their AWS resources using code.

**Implementation Recommendations**

1. Identify the Scope of Your Infrastructure Automation.

2. Design Your CloudFormation Templates.

3. Test and Validate Your Templates.

4. Deploy and Manage Your Infrastructure.

* + 1. Monitoring and Logging

**Monitoring Metrics**

AWS Serverless Application Repository is integrated with AWS CloudWatch, which is a service that provides a record of actions taken by a user, role, or an AWS service in the AWS Serverless Application Repository. And can view near-real-time analysis of data that identifies anomalies, indicators of compromise, performance issues, and configuration changes. CloudWatch captures all API calls for the AWS Serverless Application Repository as events. The calls captured include calls from the AWS Serverless Application Repository console and code calls to the AWS Serverless Application Repository API operations.

**Alerting**

Serverless containers logs are shipped to CloudWatch log group metrics, from there we can visualize and build the below use case alerting rules.

* Unusual function durations
* Real-time application and system monitoring
* Timeouts
* Long-term log retention
* Out-of-memory occurrences
* Escalating invocations
* New error types, and escalating error rates

**Logging Solution**

We recommend using Amazon CloudWatch Logs for logging in Serverless. CloudWatch Logs provides a centralized logging solution that can help you to collect, monitor, and analyze logs from your Serverless containers and applications.

Supported log formats include Amazon VPC Flow Logs, AWS CloudTrail, AWS Lambda, Common Log Format, Space Delimited, JSON, Apache web server logs, and other (user-defined) formats.

* + 1. Disaster recovery

Serverless services like Lambda, API Gateway, SQS, and SNS all automatically span across all availability zones in each region. This means you don’t have to worry about spinning up multiple instances in a multi-AZ architecture because AWS handles it for you.

You get high availability and automatic redundancy within a single region out of the box. Load-balanced, scaled architecture is part of the managed service, so you don’t have to worry about hitting your RTO.

## Technology Stack – Indicative

|  |  |
| --- | --- |
| **Feature** | **Technology** |
| CSMS User Interface | ReactJS, Redux, D3JS |
| Cloud | AWS |
| API Gateway | AWS API Gateway |
| Microservices / Scripting | NodeJS, Python |
| Unit Testing framework | Jest |
| Messaging Queue | SQS, SNS |
| Authentication | MSIL Active Directory, JWT, AWS Cognito, AWS Lambda |
| Database / Storage | MongoDB, AWS S3 |
| Mailer | AWS SES |
| DevOps | CodeCommit, CodeBuild, CodePipeline, CodeDeploy, , Lambda, S3, SonarQube, CloudFormation, CloudWatch |
| API Testing (Manual) | Postman |
| API Automation | Rest assured |
| ETL Jobs | AWS Glue |

# Training

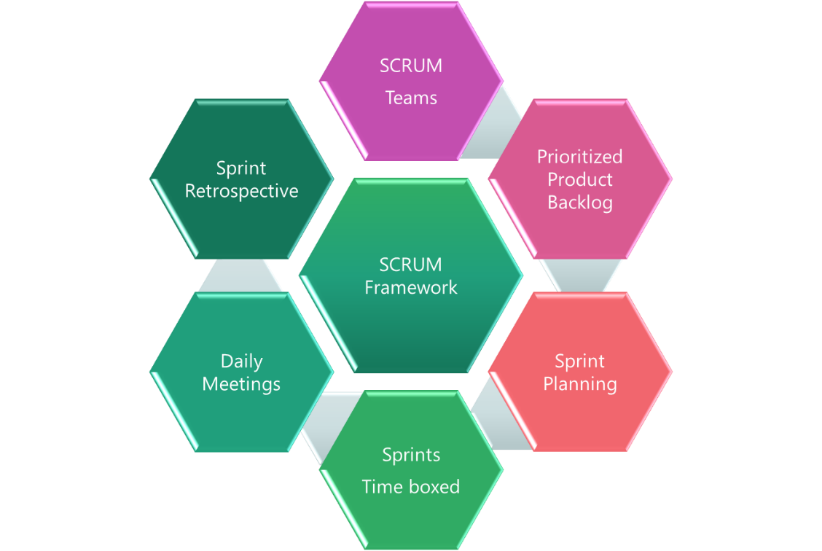
Happiest minds will provide one time training to the identified MSIL employee(s) on smart spares platform covering all the features. The training will be online training (remote) and sessions can be recorded to use for any future training purpose.

We expect 3 sessions to 2 hours each would be sufficient to cover all the features of the smart spares platform and the queries related to the platform.

# Project Delivery Approach

Agile Scrum framework would be used for project execution. Scrum ensures transparency in communication and creates an environment of continuous progress. The below diagram depicts the model followed by Happiest Minds in Agile projects that helps in

.



2-week sprint cadence would be followed and there would be a demo at the end of each sprint.

**Graphical user interface, application

Description automatically generated**

**Happiest Minds proposes Sprint based delivery by following Sprint ahead approach as outlined below.**

****

* Product Team to stay 2 weeks ahead of Solutions/Design Team & the Solutions/Design Team to stay 2 weeks ahead of development team
* Product team to keep enough prioritized stories to maintain healthy backlog supporting 2 upcoming groomed sprints
* Design teams to keep the deliverables ready for development team to pickup for the upcoming sprints
* Leverage the lead time for system design and resolving dependencies. Architects, Tech Leads, Scrum Masters to get involved

# Governance Model

Text

Description automatically generated

## Project Communication Model

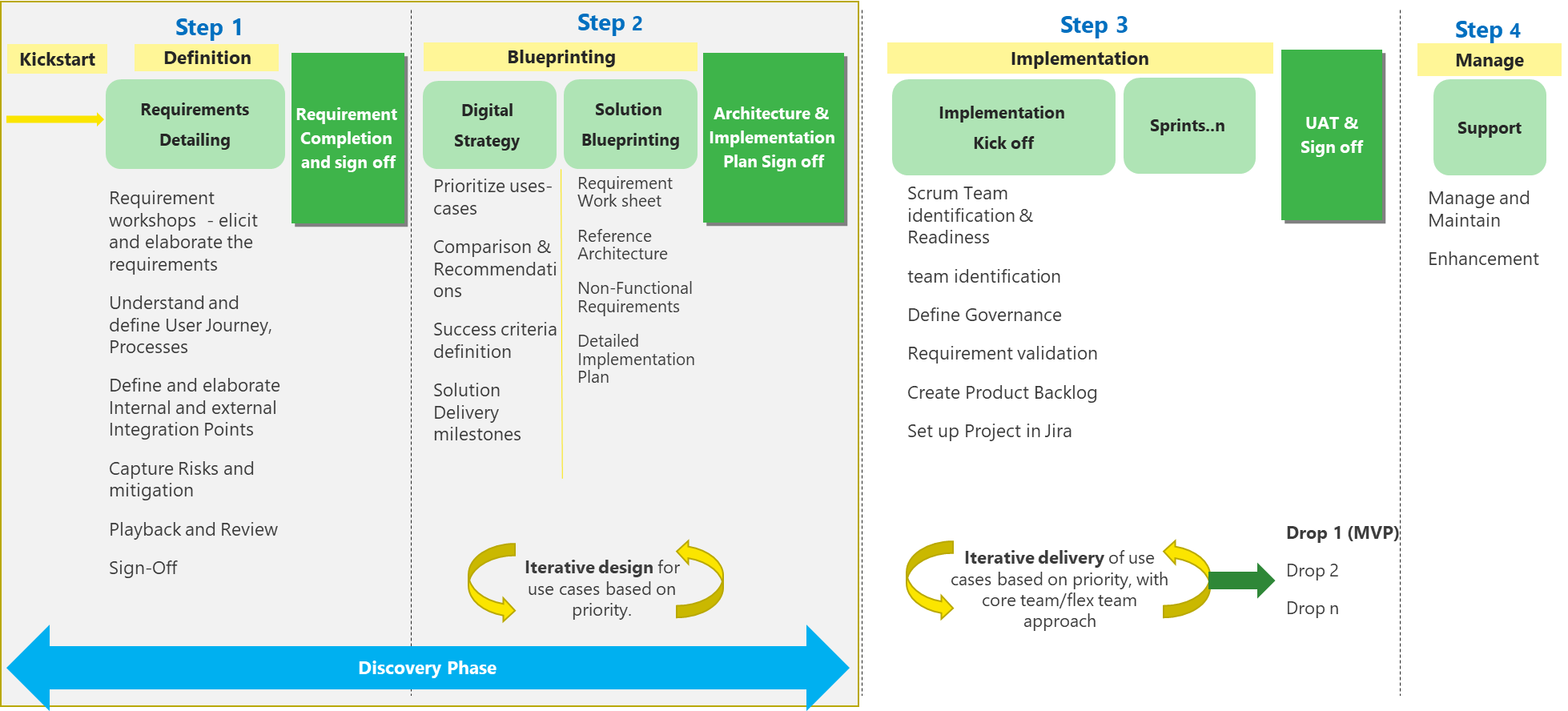
The communication model will involve the below activities for successful delivery of the project:

1. **Daily Stand-up meetings**: Between Offshore team and Project manager to discuss tasks completed, task planned and any outstanding blocking issues. MSIL is not required to participate in these meetings, but they are welcome to participate if they want to.
2. **Weekly Status Meeting**: Between Project manager at Happiest Minds and Project manager of MSIL to discuss weekly status report and next week’s plan, issues, risk on the on-going phase of the project.
3. **Monthly Operational Meeting**: Between Delivery Manager and Project Manager at Happiest Minds and Project sponsor at MSIL to discuss progress on the project, issues, and risk with resolution for project delivery and commercials. Monthly meetings can be planned if there are plans to continue further enhancements of this MVP.

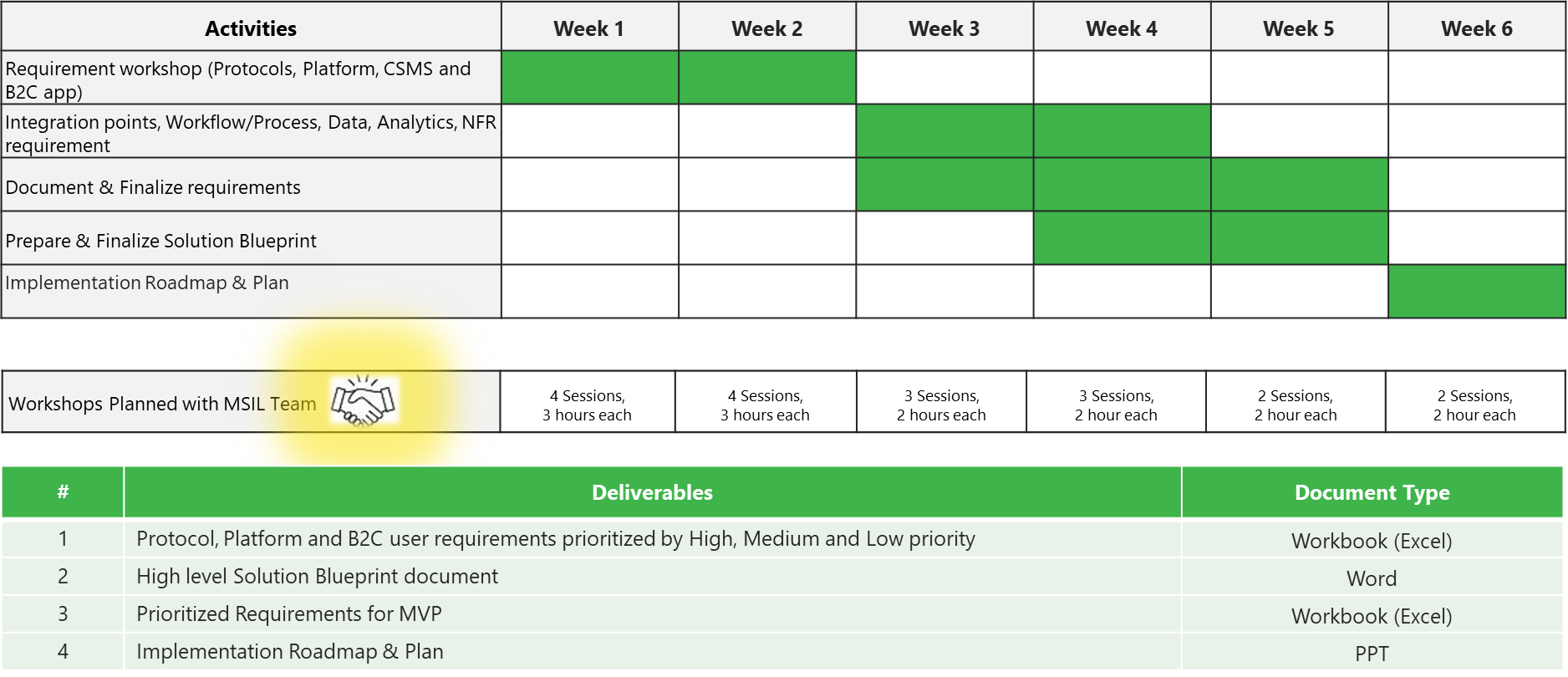
## Execution Schedule and Deliverables

Happiest Minds proposes the project execution starting with a detailed Discovery phase followed by phase-wise MVP cycle. Happiest Minds proposes a Time and Material based approach with a Core-Flex model to manage teams’ capacity as per the prioritization and capacity planning done for each of the MVPs during Discovery phase. The Core-Flex model will provide flexibility to have better control on cost, timeline and quality of the project and deliverables.

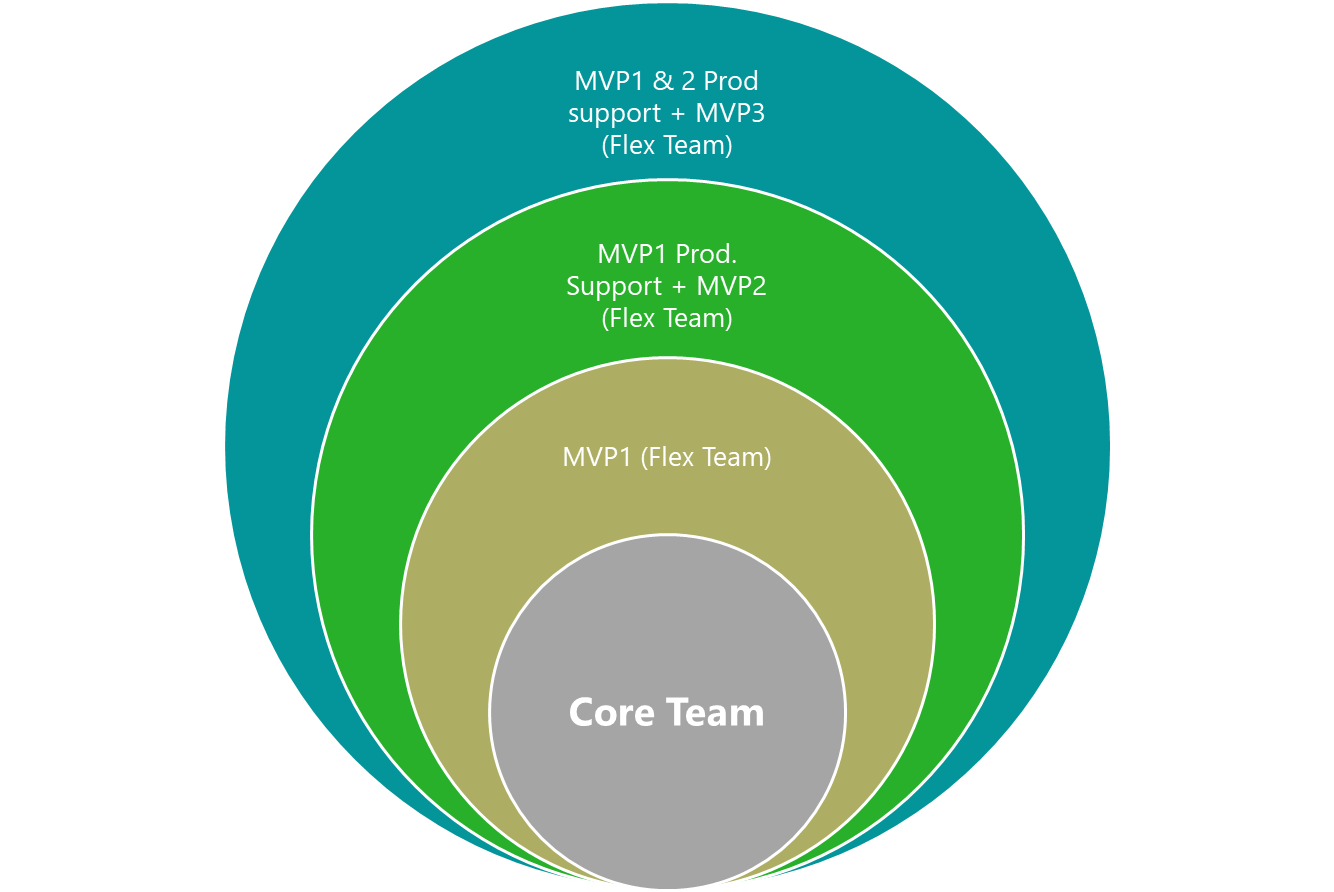
**Overall Execution Approach**



**Discovery Phase Plan**

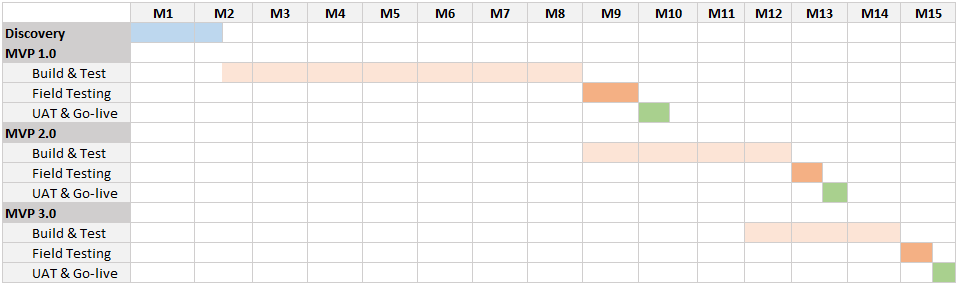


**Core-Flex model**



1. Identify Core Team with key competencies needed at the start and throughout the project lifecycle. This core team would comprise of the Product Team and Solution/Design Team along with key competencies needed from start to end of the project lifecycle.
2. Run the Discovery phase with Core team to detail out the features and options and finalize the platform requirements and architecture.
3. Identify and plan the Flex team needed to delivery subsequent phases of MVP1, MVP2 and MVP3 as well as carry out post-production support after Go Live
4. Execute the project with Sprint ahead approach.
5. At the end of each MVPs, Identify the flex team needed to support post-production issues and Business requests.

**Below are the tentative timelines for the program, we will work jointly with MSIL and finalize this by end of the Discovery Phase.**



## Risk and Mitigation Plan

| **No** | **Risk** | **Mitigation** |
| --- | --- | --- |
| 1 | Data lake should be available start of the project to build this system | Manual feeds need to be loaded to table temporarly. |
| 2 |  |  |

## Change Request Management

Changes to the scope will mean any of the following:

* Change introduced in the application between the proposal and the actual implementation or during the implementation.
* Any changes to the scope of the project as detailed in section “In Scope”
* Invalidation of any of the assumptions detailed in section “Assumptions”
* Any change to the terms and conditions as defined in section “Commercials”
* Non-fulfillment of any of the dependencies detailed in the sections “Dependencies”
* Any delay that happens to the schedule can be attributed to MSIL.

In case of a change request, the scheduled end date for this Project and/or the fees associated may change. Whenever a change is identified, it will be managed as per the below process:

* For any changes to the scope, either MSIL or Happiest Minds will submit a Change Request
* Happiest Minds will issue a Change Order providing the impact of the change to the schedule and/or fees
* MSIL SPOC will review along with Happiest Minds SPOC and mutually agree to either approve or cancel the change order
* Changes will be implemented only after MSIL SPOC approval and signing of the change order form by both the Parties
* For any MSIL dependencies that are not met or issues that are not resolved, which could impact the schedule – Happiest Minds Project Manager will complete a Change Order and inform the MSIL SPOC.

## Acceptance Criteria

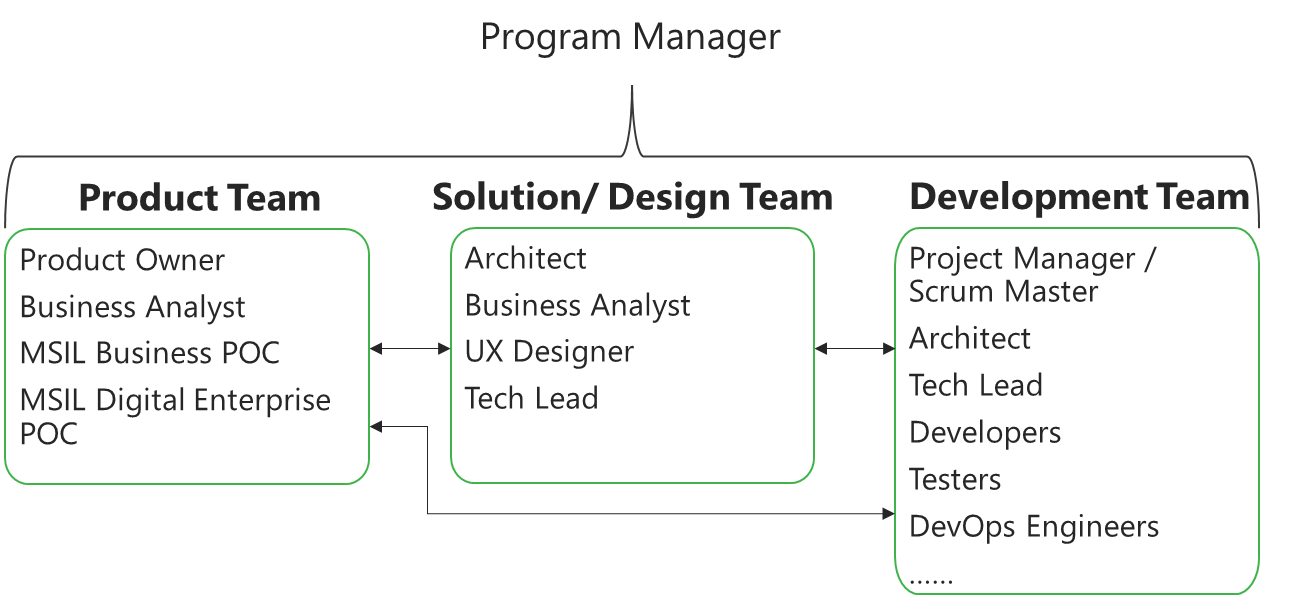
* The user story’s acceptance criteria/test cases will be reviewed and mutually agreed upon.
* The Acceptance test case document along with acceptance test environment details would be shared by MSIL and signed off 4 weeks before the start of Acceptance Testing
* Acceptance Testing will be carried out by MSIL team and supported by happiest minds
* Acceptance Testing should be completed within 2 weeks from the delivery of the system tested application
* The acceptance criteria will be passing of the user acceptance test cases with Zero Critical (P0) and High (P1) severity bugs identified

**Defect Severity – Definition:** Definition of Defect Severity and Priority are as below.

|  |  |
| --- | --- |
| **Defect Severity** | **Definition** |
| P0 – Critical | Defect may be a showstopper – that is, it stops the user from using the system further. |
| P1 – High | Defect occurs repeatedly and prevent the user from proceeding in the normal way, but a workaround exists. |
| P2 – Medium | A defect is isolated or does not stop the user from proceeding but is annoying and causing inconvenience. |
| P3 – Low | A defect that in no way affects the performance or functionality. E.g.: Aesthetic issues and grammatical errors in messages. |

* The deliverables would be deemed accepted if there is no response/feedback from MSIL within 2 weeks post release of System Tested Application.

# Proposed Team Structure



## Key Team Members (Indicative)

|  |  |
| --- | --- |
| **Project roles** | **Count** |
| Program Manager | 1 |
| Platform Architect | 1 |
| Backend Technical Lead | 1 |
| Senior Backend developer | 3 |
| UI Technical Lead | 1 |
| Senior UI Developer | 2 |
| DevOps Lead | 1 |
| Mobile Lead | 1 |
| Performance Testing engineer\* | 1 |
| Test Lead | 1 |
| Senior Test engineers (automation) | 2 |
| Python Lead | 1 |
| Python Engineer | 2 |

*\*These roles will be onboarded based on the project schedule or as needed during the engagement*

# Commercial

|  |  |  |
| --- | --- | --- |
| **Sr. No** | **Project roles** | **Rate Card/day in INR** |
| 1 | Program Manager | 27000 |
| 2 | Domain Consultant | 30000 |
| 3 | Platform Architect | 35000 |
| 4 | Backend Technical Lead | 17750 |
| 5 | Senior Backend developer | 15500 |
| 6 | UI Technical Lead | 17750 |
| 7 | Senior UI Developer | 15500 |
| 8 | DevOps Lead | 15500 |
| 9 | Mobile Lead | 17500 |
| 10 | Senior Mobile Developer | 15500 |
| 11 | Performance Testing engineer\* | 12000 |
| 12 | Test Lead | 12000 |
| 13 | Senior Test engineers (automation) | 12000 |

* Travel: Any travel and boarding/lodging cost for requirement discussion, field testing or production release would be taken care by MSIL or it would be charged at actuals to MSIL .Happiest Minds will seek prior permission from MSIL before undertaking any such trip.
* Project specific Costs and Expenses: Happiest Minds will provide standard Microsoft Windows based PC hardware and software to its team members at its site for execution of work under this project. Any project specific specialized hardware, software licenses, testing devices or networking/cloud infrastructure required for the project will either be provided by MSIL or will be procured and expensed to MSIL. Happiest Minds will obtain prior written approval from MSIL before procuring or incurring any project specific hardware, software, devices, or network infrastructure expenditure.
* Taxes: The pricing mentioned excludes GST and any other local and country specific taxes including any withholding tax, as may be applicable.
* Invoicing Schedule: The invoicing would be done on basis on the timesheet submission on monthly basis for this project.
* Payment Term: Payment of invoices shall be in accordance with the terms of the Master Services Agreement

# Abbreviations: