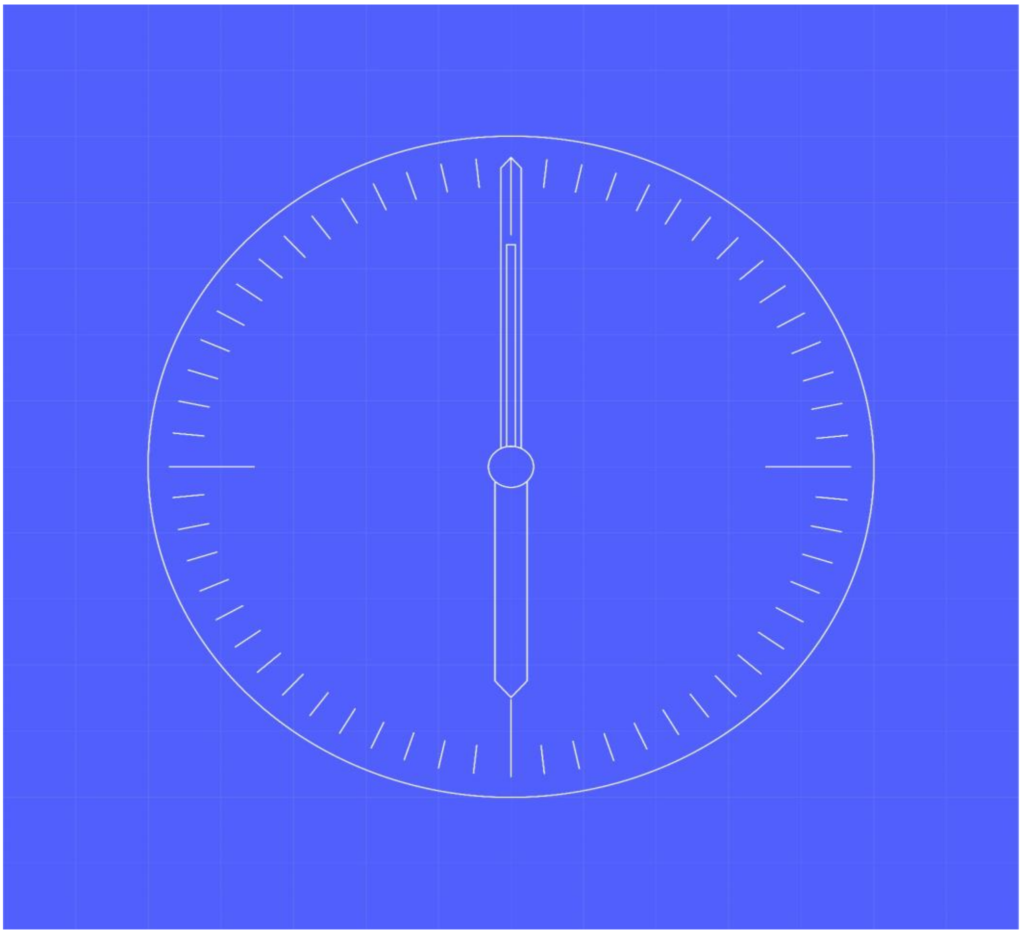
MHHS Programme

Release and Configuration Approach and Plan



Document owner	Document number	Version
SI Workstream Lead	**MHHS-DEL1089**	**1.~~0~~4**
Status:	Date	Classification
Approved	~~17 May, 2023~~8 July, 2024	**Public**

1. Contents

Figures

Tables

1.1 Change Record

Date	Author(s)	Version	Change Detail
27 January 2023	Simon Berry	0.1	Initial Draft for LDP peer review
7 February 2023	Simon Berry	0.2	Draft for SRO review
10 March 2023	Simon Berry	0.3	Draft for Core Systems providers review
10 May 2023	Simon Berry	0.4	Updates following Core Systems providers review
17 May 2023	Simon Berry	1.0	Document baselined following approval at ~~TMAG~~SITAG on 17th May
1 March 2024	Simon Berry / Sreeja Dutta	1.1	Document updated to reflect revised release schedule and housekeeping changes
22 March 2024	Simon Berry / Sreeja Dutta	1.2	Updates following SRO review
26 June 2024	Simon Berry / Sreeja Dutta	1.3	Updates following Qualification team review
8 July 2024	Simon Berry / Sreeja Dutta	1.4	Updates following SRO review

1.2 Reviewers

Reviewer	Role
~~Adrian Page~~	~~LDP SI Workstream Lead~~
~~Kate Goodman~~	~~LDP Test Architect~~
Paul Pettitt	LDP Design Assurance Lead
~~Nigel Hunt~~Lee Cox	LDP SI Test Team
Dominic Mooney	LDP SI Test Team
Adrian Ackroyd	SRO Function Programme Test Manager
Chris Welby	MHHS SRO
Smitha Pichrikat	SRO Function Client Delivery Manager
Kiran Raj	SRO SIT Functional Test Lead

1.3 References

Ref No.	Document/Link	Publisher	Published	Additional Information
REF-01	MHHS -DEL 315 E2E Testing & Integration Strategy	SI Testing	~~29th~~29 April, 2022	
REF-02	~~MHHS-DEL172 Change Control Approach~~MHHS-DEL172-Programme-Change-Control-Process-Published-v1.0	MHHS PMO	~~5th May~~3 August, 2022	
REF-03	~~MHHS -030 Programme Governance Framework~~MHHS DEL-030 MHHS Programme Governance Framework V3.2 (clean version)	MHHS PMO	~~22nd June, 2022~~6 December, 2023	
REF-04	~~MHHS-DEL813 Test Data Overarching Approach & Plan~~ MHHS-DEL813 Overarching Test Data Approach & Plan v1.0	SI Testing	~~Jan~~19 July 2023	
REF-05	~~MHHS-DEL 466 Defect Management Plan~~MHHS-DEL466 Defect Management Plan v2.0	SI Testing	~~23rd May, 2023~~29 February, 2024	
REF-06	~~MHHS DIP 094 Interface Code of Connection~~MHHS-DEL1197 - Interface Code of Connection v1.4	~~SI~~MHHS DAG	~~May 2023~~12 February 2024	~~Draft version~~
REF-~~07~~08	~~MHHS-DEL1089 Release and Configuration Approach & Plan~~MHHS-DEL618 Environments Approach & Plan v3.0	SI	~~17th May, 2023~~7 June, 2024	
REF-~~08~~09	~~MHHS-DEL618 Environments Approach & Plan~~MHHS-DEL1118 Qualification Approach and Plan v2.0	~~SI~~BSC and REC Code Bodies	~~24th February, 2023~~3 April, 2024	

Formatted Table

1.4 Terminology

Term	Description
Various	For terminology, see Programme Glossary on the MHHS portal: Programme Glossary (SharePoint.com)

2 Executive Summary

The Market-wide Half Hourly Settlement programme (MHHS) when completed will contribute to a more cost-effective electricity system, encouraging more flexible use of energy and helping consumers lower their bills. The responsibility for success is shared between all parties and stakeholders, with everyone working together to make sure the programme is delivered and in the highest possible quality.

Robust quality assurance for the necessary changes is required for this complex programme, not least during the industry testing stages where proactive and efficient management of system environments will be crucial to the overall success of the programme objectives. This document provides definition to MHHS Industry Test participants on the approach for testing environments covering planning, scheduling, management, coordination, readiness, and maintenance. The approach and process adopted is intended to be familiar for those industry test participants involved in recent industry programmes.

~~At this stage in the programme information is dependent on artefacts being available at later dates and are listed as follows:~~

- ~~[REF-04] Test Data and Overarching Approach & Plan – outlines key data requirements which can assist when assessing environment sizes~~
- ~~[REF-06] DIP 094 Interface Code of Connection – provides technical information including connection and security details~~

3 Introduction

3.1 Document Purpose

The purpose of this document is to define how Release and Configuration Management shall be undertaken within the MHHS Programme testing. The aims and objectives of the Release and Configuration Management Approach and Plan are designed to deliver, distribute and track one or more changes in a release into the test environments.

The Release and Configuration Management Approach is mandatory for all involved in Release activities within the MHHS Programme. Release and Configuration Management Processes, Procedures and Work Instructions shall not deviate from the principles laid down in the Release and Configuration Management Approach. Programme Participants are required to comply with the MHHS Test Plan in accordance with BSC Section C12. The Release Management and Configuration Management Approach forms part of the MHHS Test Plan.

The Release and Configuration Management Approach and Plan controls the release through the Deployment stage this ensures that only authorised and quality-controlled versions of software are introduced into the MHHS Test Environments.

This Approach covers scope of the Release Management Service, Release Classifications and Frequency, Release Lifecycle, Governance, Roles and Responsibilities and Toolsets in use for Release Management purposes. For clarity, this document covers the Release Approach throughout the testing of the MHHS Programme, including the SIT and Qualification phases.

This Approach is based upon the following Release Management principles:

- Release Management is a thread that provides coherence and control between the phases / stages of change to software components of the solution, across the programme.
- Release Management is closely aligned with Change Management, Incident Management, Configuration Management, Requirements Management, Environment Management and other disciplines as required.
- Releases are structured for a series of changes to the Test environments, optimised using a variety of release approaches (sequential, parallel, emergency).
- Releases are planned and prepared with the MHHS Programme and Programme Participant organisations in mind, considering all areas of impact and not just the technical change.

Releases are planned and prepared with the MHHS Programme and Programme Participant organisations in mind, considering all areas of impact and not just the technical change. Post Programme ie. following the end of the qualification test phase, the release management processes will need to be developed by Elexon in order to state how they will manage production releases. In addition, the release management processes will operate in production during migration, also need to be defined. This document does not seek to define those processes.

Development & Build, Testing and Environment Strategies which are part of Release Management Service are described in separate documentation (please refer to the “Related Documents” section of this document).

This document should be read by the following groups:

- MHHS Core Systems Providers engaged in MHHS programme design, build and industry testing activities
- SRO Function (SRO)
- Lead Delivery Partner (LDP)
- Core Programme Team (CPT)
- System Integration Team (SI)
- Programme Party Coordinator (PPC)
- Programme Management Office (PMO)
- Systems Integration Testing ~~and Migration~~ Advisory Group (~~TMAG~~SITAG)
- Qualification Advisory Group (QAG)
- Environment Working Group (EWG)
- Independent Programme Assurance (IPA)
- DIP Service Provider
- SIT Participants

3.2 Reviews and Approvals

The document will be reviewed by the following team members:

- ~~Kate Goodman~~Lee Cox, LDP SI Test ~~Architect~~Lead
- Paul Pettitt, LDP Design Assurance Lead
- ~~Nigel Hunt, LDP SI Test Team~~
- Nicola Farley, Non SIT LDSO Qualification Test Manager
- RECCo
- BSC
- Dominic Mooney, LDP SI Test Team.

Upon completion of LDP/Expleo review, it will then go through a formal SRO team review by:

- Adrian Ackroyd, SRO Function Programme Test Manager
- Smitha Pichrikat, SRO Function Client Delivery Manager
- Chris Welby, MHHS SRO~~.~~
- Kiran Raj, SRO SIT Functional Test Lead

Upon completion of the SRO review it will then be distributed to the EWG for review where comments will be incorporated leading to a recommendation of approval by the group.

When comments and feedback have been incorporated, approval will be requested from:

- Systems Integration Testing ~~and Migration~~ Advisory Group (SITAG) (Previously known as TMAG~~).~~)

The document will also be reviewed by Programme Participants before submitting to the ~~TMAG~~SITAG and will be made available for information via the programme portal.

3.3 Change Forecast

This document will be reviewed and where applicable, updated when the following are available or updated:

- ~~Baselined E2E MHHS Design~~
- Re-baselined MHHS Programme Plan
- ~~MHHS Data Approach Plan~~
- ~~DIP Design implementation.~~

The SI Test Team will own this document and maintain it, with review and approval by MHHS programme governance as appropriate. Each new version supersedes the previous version in its entirety. It will follow the Programme's change control process governed by the PMO [REF 02] MHHS-DEL172 Change Control Approach

Updates to this document will follow the review and approval process outlined in section 3.2.

3.4 Summary of Changes

This is version 1.~~0~~3 and quality checks have been undertaken by the author and the LDP and SRO reviews.

4 Objectives

4.1 Key Points

The objective of the MHSS Release Management & Configuration Management Approach and Plan is to ensure that there is coverage for the planning, scheduling and governance of MHSS Releases into the test environments provided by the Central Parties (DCC, Electralink, Elexon, Recco, St. Clements, ~~DIP Provider~~Avanade) that are required in order to support MHHS Programme Testing. The environments remain under the governance and control of the respective Central Parties, however Releases to those environments are required to conform to the MHHS Programme Release Management Approach.

The Release & Configuration Management service covers changes to the infrastructure and environments (including data) onto which the MHHS Programme software is deployed in order to ensure that the configuration of the overall solution is in a known state for the test execution phases of the MHSS Programme.

The primary document audience is Central Parties and Programme Participants as they are the parties that need to undertake activity to deploy a Release. ~~The~~Although the process is slightly different for Central Parties to general Programme Participants~~ are 'accessors' of a Release rather than a party that implements a Release~~.

The process and mechanisms within this document fully support the underlying principles described in ~~[REF-01] MHHS E2E Testing & Integration Strategy and in [REF-08] MHHS-DEL618 - Environment Approach Plan v2.2.docx~~[REF-01] MHHS E2E Testing & Integration Strategy and in [REF-08] MHHS-DEL618 - Environment Approach Plan v3.0.

4.2 Assumptions and Caveats

This document is written to aid Release and Configuration Management for Deployment Planning~~. However, there are certain caveats which will hinder early versions of this document due to the unavailability of key information. Therefore, the intention is to add more detailed information as when this information becomes available. This should not impact approval of the initial publication of this document where the assumption is that approval is based on the intended content.~~ in SIT and Qualification test phases.

~~Key information include:~~

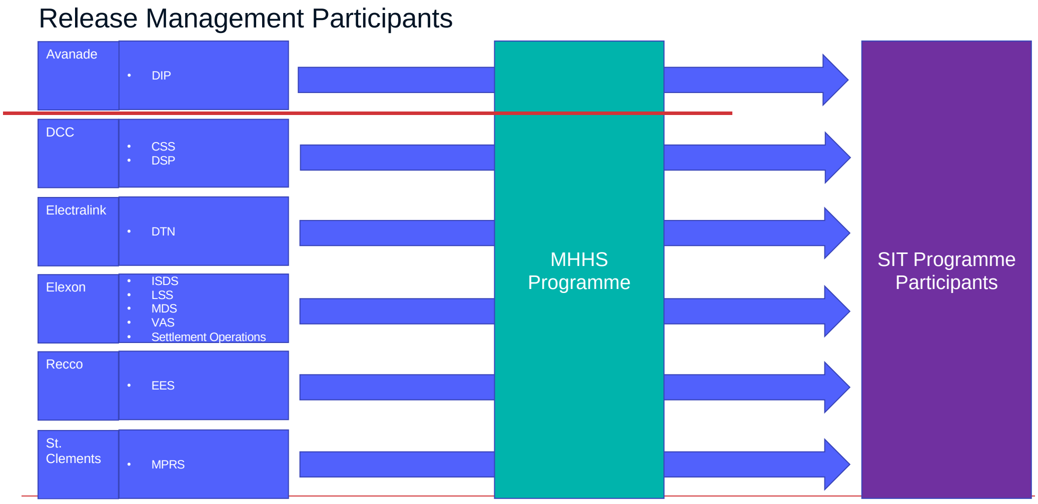
- ~~Baseline dates and guidance of the programme.~~
- ~~Scheduling of tranches based on Test Plans.~~
- ~~Core System Provider Release Management Processes & Procedures~~

4.3 Environment Working Group (EWG)

- The MHHS Environment Manager will work with the EWG to initially shape the Release and Configuration Management Approach & Plan. The MHHS Environment Manager is responsible for defining the Release & Configuration Management Approach and ensuring that the processes and procedures defined are adhered to. Where appropriate, the MHHS Environment manager will report these activities back to the EWG, Programme Participants and Stakeholders including QWG.
- The EWG will report their output to the ~~TMAG~~SITAG for approval. This will occur on an ongoing basis and may require engagement with other programme participants. Where the EWG is unable to reach a consensus on a decision delegated to them by ~~TMAG~~SITAG the matter will be escalated to the ~~TMAG.~~ SITAG. If required, QAG to be informed of the decision made at SITAG.

5 Scope

5.1 Release Management Participants



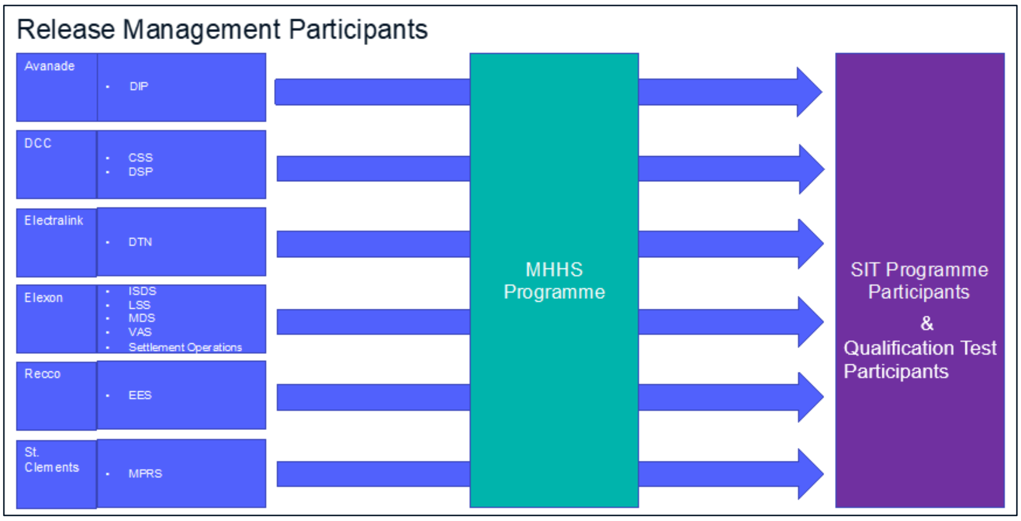


Figure 1 Release Management Participants

The processes defined in this document govern the Central Parties and Core ~~Systems Providers + ~~Solution Provider (St. Clements) releases that will be coordinated by the MHHS Programme.

The processes defined in this document apply up to the MHHS Programme Milestone, ~~M11~~M14. The transition arrangements will define the release approach in Production for the period where Migration is under way and MHHS Programme Qualification testing is still ongoing. This period will be circa April 2025 to January 2026.

5.2 Out of Scope

- Design, Build & Development
- DIP Provider and Core Systems Provider PIT Testing phases
- Environments
- Data
- Production Release & Configuration Management - either the MHHS Programme Deployment Planning or Service Management Approaches will define ~~how~~ the Release Approach post-~~M11~~M14.

5.3 Release and Configuration Managers

Release and Configuration Managers play a key role in ensuring that code is released into environments successfully. The scope of this document will try to address the expectations for members of the Environment Working Group (EWG) and that of Release and Configuration Managers.

These will be covered comprehensively in ~~the Roles and Responsibilities section~~Section 14, Roles & Responsibilities.

5.4 Participants

All Test Participants involved in MHHS testing will be expected to comply with the Release and Configuration Approach & Plan.

These are:

- Suppliers
- ~~Service providers~~
- Agents
- DIP Connection Providers (DCP)
- 3rd Party IT Organisations supporting MHHS Programme Participants
- (I)DNOs
- Avanade, Data Integration Platform service provider (responsible for the DIP)
- DCC (responsible for both Smart Metering and Central Switching Service)
- Elexon (responsible for Elexon Central Systems, which comprise Load Shaping Service, Market Data Service, Volume Allocation Service, Industry Standing Data Service and BSC Settlement Operations)
- Electralink (responsible for the Data Transfer Network – DTN)
- St Clements and C&C, together with the (I)DNOs (responsible for ~~SMRS~~Registration Services)
- UMSOs (responsible for the UMSO services)
- RECCo (responsible for EES).

Each of the parties above is referred to as a Test Participant (TP) throughout this document.

5.5 Test Phases

There are various test phases which determine how and when Test Participants will execute testing. These test phases can be referenced in the [REF-01] MHHS-DEL315 - E2E - Testing & Integration Strategy document. ~~The SI Test Team will schedule the test phases and work with the MHHS Environment Manager for Release & Configuration Management coordination and planning.~~

- SIT
- Non SIT LDSO Qualification Testing
- Non SIT Supplier & Agent Testing

Formatted: French (France)

Formatted: Indent: Left: 1.27 cm

5.6 Coordination and Planning

Planning and scheduling of testing for each Test Participant will be outlined in the Programme's central test plans. This will align with the scheduling for deploying Releases into the test environments.

6 Management and Coordination

6.1 Planning

SI Release & Configuration Management Team will advise Test Participants when code releases~~, Test Data or other configuration items will be deployed to the SIT and UIT test environments~~.

~~In further drafts, this section will contain a high-level plan of the testing schedule which will be based on the re-plan.~~

6.2 Tracking and Coordination

- The MHHS Environment Manager and MHHS Release Manager are central to all coordination, communication, and escalation.
- EWG will have a regular meeting scheduled on the first Tuesday of every month. This will be chaired by the MHHS Environment Manager. Test Participant Environment Managers, Release Managers or representatives are encouraged to attend as this will be particularly important during both preparation and execution of SIT and UIT phases. Programme test leads and representatives from the design groups are also invited to attend.
- As the project progresses, there may be a need for more frequent meetings covering issues or testing progress. Only interested parties need attend these meetings or 'catch-ups'.
- ~~The MHHS Environment Manager will produce a high-level testing and readiness report at regular intervals. Most likely, these will be weekly, but could increase in frequency based on activity.~~
- Any scheduling of test phases will be shared via a centralised Gantt chart (or similar) which will be visible to the EWG, and other interested parties published via the MHHS collaboration base.
- ~~ADO Dashboards will be utilised to track versions of components in environments. This will be part of the release management process which will be referenced here when it is defined. Dashboards will allow visuals of release versions currently in environments and the ability to drilldown into historical versions. Owners of Central and other components critical to the end-to-end settlement processes are expected to communicate their current application versions via release notes as this will avoid confusion when testing on correct versions of various system components. Critical areas will be identified by the Programme and listed here. Programme will decide if this is required for non-critical systems.~~
- ~~A release management plan and schedule will be published to inform when fix versions will be applied to environments. This will be published on the MHHS collaboration base.~~
- ~~Subject to the design, it is assumed the status of the programme environments will be implemented via a dashboard on a chosen tool, to ensure environment statuses (i.e., where components are available or unavailable) are fully tracked. It is currently assumed this will also cover services to and from the central systems. It should be noted that the Environments Approach & Plan document is the authority on Environment monitoring, not the Release Management and Configuration Management Approach.~~
- Azure Dev Ops (ADO) will be utilised to capture testing issues. Release and Configuration issues will have its own category and will be triaged as part of the defect workflow. Release and Configuration issues will be coordinated by the MHHS Environment Manager / MHHS Release Manager. For clarification, the defect workflow is captured in the [REF-05] MHHS-~~DEL46~~DEL466 Defect Management Plan.

6.3 Communications and Meetings

6.3.1 Mail and mail ~~groups~~group.

- EWG members will be part of the EWG mailing list for the monthly meetings.

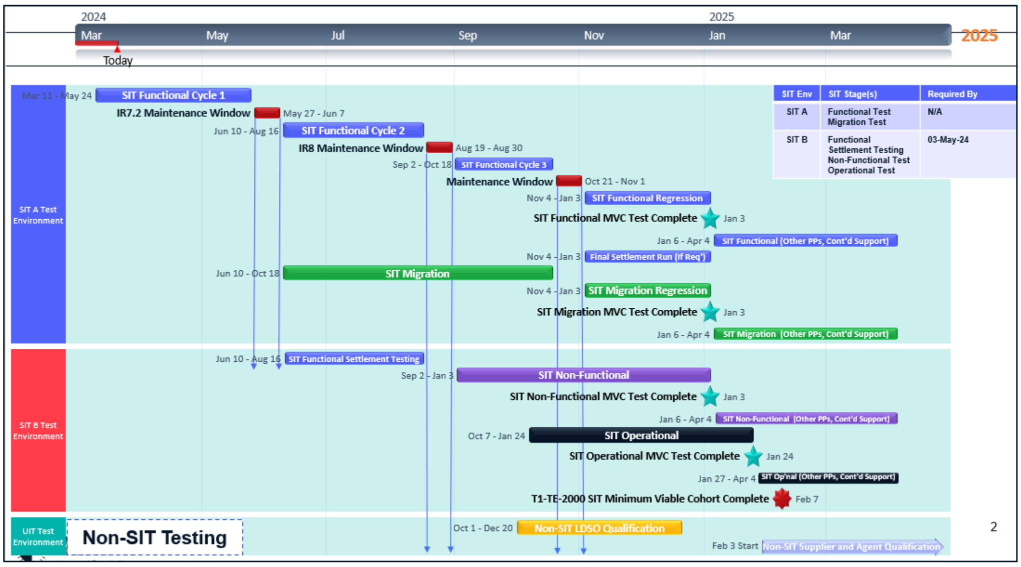
6.3.2 Meetings

Regular meetings will take place and will focus on members of the EWG and those with specific interests in the preparation and status of the test environments. Meetings will take the form of the following:

- Monthly EWG catch up will occur on the first Tuesday of every month. This is an open forum which EWG members and interested parties are encouraged to attend. It will cover open topics, key points over the past and upcoming monthly period, and any general queries that the group may have relating to Environments and Release & Configuration Management issues.

7 Release Roadmap

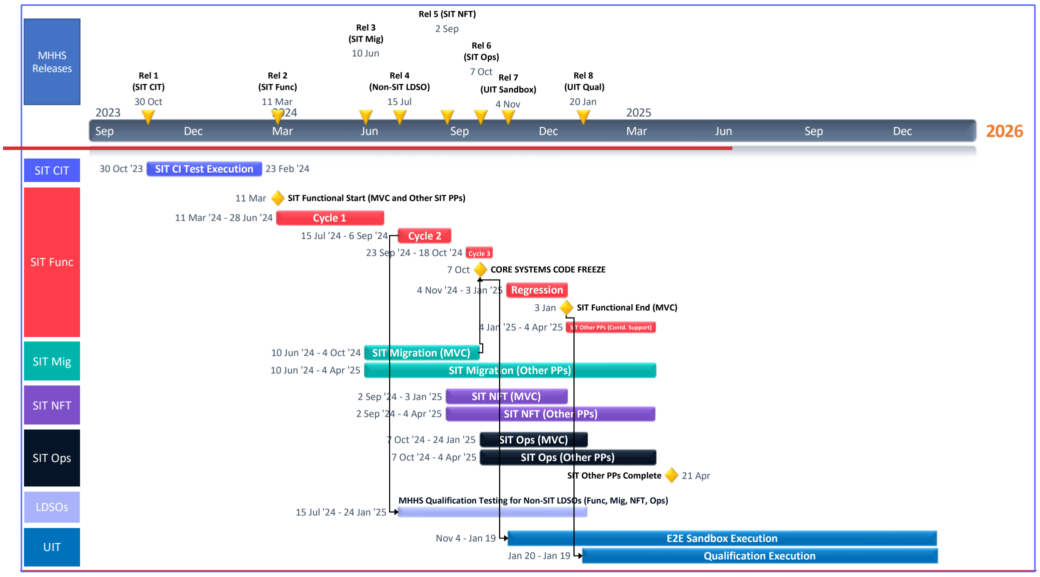
7.1 Testing Phases POAP



SIT Env	SIT Stage(s)	Required By
SIT A	Functional Test Migration Test	N/A
SIT B	Functional Settlement Testing Non-Functional Test Operational Test	03-May-24

Figure ~~2~~2 : MHHS SIT POAP

Figure 3, below states the timelines of the different test phases of the Programme.~~ A ‘Potential Releases’ swim lane has been added for illustration purposes only in order to highlight when there may be Major Releases of Code. The POAP reflects~~ and the ~~Programme Test Planning assumptions at the point of Round 3 Consultation~~major programme releases.



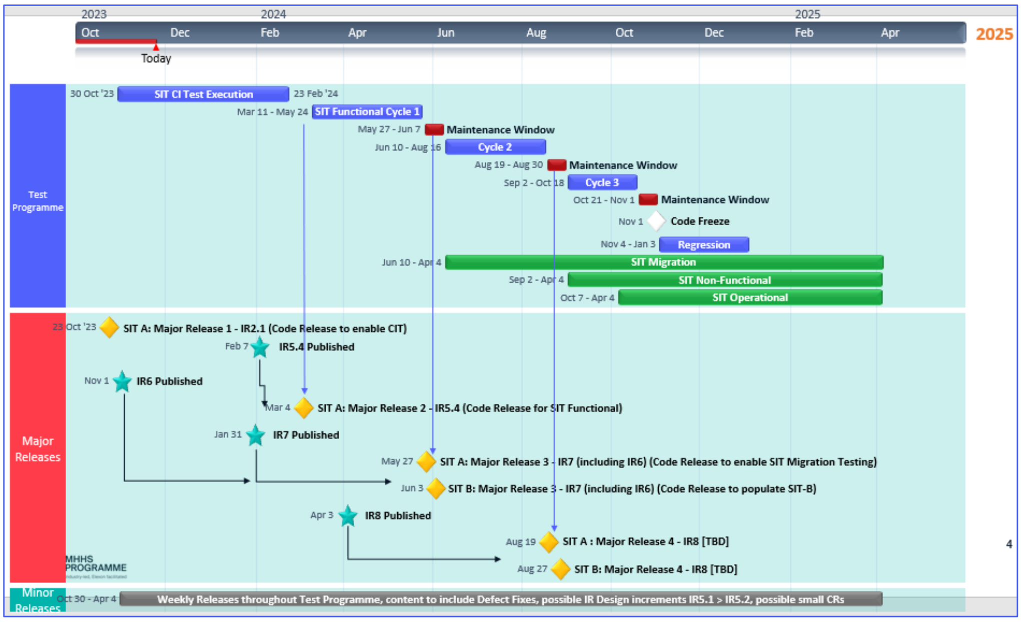


Figure 3 Testing Phases POAP with potential Releases Swim Lane

Formatted: Not Highlight

~~Potential~~

7.2 Major Releases

~~At present there is no defined Release Roadmap, however we can make some assumptions regarding the Programme events when there may be Major Releases.~~

Release	Date	Purpose of Release
~~Release~~ IR2.1	~~30~~23 October, 2023	Code Release to enable the **CIT Test phase**.
~~Release 2~~IR5.4	~~11~~4 March, 2024	Code Release to enable the **SIT Functional Test phase**.
~~Release 3~~IR7 (SIT A)	~~10 June~~27 May, 2024	Code Release to enable the **SIT Migration Test phase**.
~~Release 4~~IR7 (SIT B)	~~15 July~~3 June, 2024	Code Release to ~~enable~~populate the ~~**non-**~~**SIT** ~~**LDSO Test phase**.~~ **-B**
~~Release 5~~IR8 (SIT A)	~~2 September~~19 August, 2024 [TBD]	Code Release to enable the **SIT** ~~**NFT Test phase**.~~**Functional Cycle 3**
~~Release 6~~IR8 (SIT B)	~~7 October~~27 August, 2024 [TBD]	Code Release to enable the **SIT** ~~**Operational Test phase**.~~**Non-Functional Cycle 3**
~~Release 7~~	~~4 November, 2024~~	~~Code Release to enable the **UIT Sandbox phase**.~~
~~Release 8~~	~~20 January, 2025~~	~~Code Release to enable the **UIT Qualification Test phase**.~~

Table 1 ~~Potential~~ Major Release dates and rationale

Formatted Table

Formatted: ui-provider

Formatted: ui-provider

Formatted: ui-provider, Font color: Text 1

Formatted: Font: Bold

7.3 Minor, Patch and Emergency Releases

In addition to any Major Releases there will inevitably be many Minor Releases, Patch Releases and Emergency Releases in order to promote defect fixes and configuration changes into the various test environments to support the various programme test phases. Further detail is provided in Section 10.1, Release Types.

7.4 Alignment of SIT-A and SIT-B

Once SIT Settlement testing commences, the Programme will have SIT Functional and Migration Test Execution running in SIT-A, in parallel with SIT Settlement Test Execution running in SIT-B and SIT non-Functional and SIT Operational testing to follow in SIT B.

Design Release/Build equivalence needs to be maintained across SIT-A and SIT-B noting synchronisation cannot be achieved 100% of the time. In this situation, the following release would realign previous defect fixes.

A single Code branch approach will be maintained in support of SIT Test Execution across both Test Environments:

- Participants are expected to have a single Staging Environment to PIT test and deploy fixes from, for either SIT-A Defects and/or SIT-B Defects.
- Weekly Releases to both SIT-A and SIT-B need to be scheduled.
- An indicative Weekly Minor Release Schedule would be Monday PM to the appropriate environment(s). However, this is purely indicative and defect fixes to environments whilst needing to be timely also need to consider the impact of any environment downtime.
- Defect Retesting will reside solely in the SIT Test Environment that uncovered the Defect e.g:
  - Weekly/Minor Release 5.2.16 contains:
    - Defect 123, Defect 456, Defect 789 for SIT-A
    - Defect 246, Defect 468 for SIT-B
  - All 5 Defect fixes are deployed into SIT-A and SIT-B within the same working day.
  - SIT-A SIT Functional Test responsible for - Retesting Defect 123, Defect 456, Defect 789
  - SIT-B SIT Settlement Test responsible for - Retesting Defect 246, Defect 468

7.5 Confirming Interim Release Implementation Dates

IR7.1 and IR8 have defined implementation dates aligned to milestones in the testing POAP (See figure 3).

There may be other IR releases i.e. IR5.5 where the implementation does not necessarily align to testing milestones. For these releases, the SI Release team will work with Avanade, SRO, SI Test team and SI Design team in order to determine if there is a logical sequence in which Central Parties and Programme Participants need to implement the release. The SI Release team will then work to determine an appropriate implementation date.

Formatted: Font: 10 pt

8 MHHS Environment Overview

8.1 Path from Development to SIT / UIT Environments

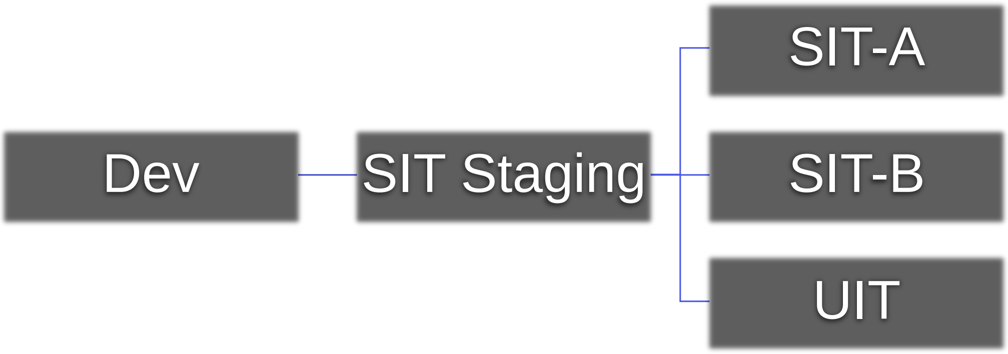


Figure 4 Path to Deploying Code from the Development Environment to Test Environments

This diagram articulates the ideal path to deploy code to the Test Environments. A Central Party would develop code in a Development environment and then deploy to SIT-A, SIT-B, UIT environments as appropriate. If the defect was identified in the UIT environment and has not previously been encountered in the SIT environment(s) it is expected that the defect fix would be deployed to the UIT environment first and then the SIT environments at the next opportunity so as not to impact testing velocity.

It should be noted that whilst the preference is for Core Systems Providers to utilise a Staging Environment it is recognised that not all will do so.

8.2 Environment Overview

The following table is the assumed usage strategy for each central system environment required for test phases. Whilst it is the Programme preference that Central Parties have a SIT Staging Environment it is not mandatory. It should also be noted that some Central Parties may not provide separate environments for SIT-A, SIT-B and UIT.

The actual timelines will be agreed with the Programme and a reference will be added here when that is available:

Environment	Phase	Testing Stage	Comments
SIT Staging			Readiness for SIT such as regression for changes, defect re-testing, etc. This will ensure that the actual SIT environments are not broken when new code is deployed.
SIT A	SIT	SIT Component Integration SIT Functional SIT Migration SIT Functional Regression	Component integration tests will be conducted as individual components are integrated. Then full end-to-end testing can start.
SIT B	SIT	~~SIT Migration~~ SIT Non-Functional* SIT Operational SIT Functional Settlement Testing SIT Non-Functional Regression	It is assumed these three stages can be executed on one environment, but not in parallel to avoid conflicts. TP's can decide to have their own environment for each stage or re-purpose their environments for each stage. *Note new systems, such as the DIP, may be required to run tests on Pre-Prod and Prod.
UIT	UIT	Qualification E2E Sandbox	~~Central systems~~The DIP and ~~some (I)DNOs' environments~~ two test harnesses, one for non-SIT LDSO Qualification Testing and one for Non-SIT Supplier & Agent Qualification Testing will be provided as a testing service to allow ~~TPs~~Test Particpants to conduct Qualification Testing and ~~E2ESandbox Testing. Each TP will need to complete either SIT or Qualification Testing before starting~~ E2E Sandbox Testing.

Table 2 Environments per Test Stage

Formatted: French (France)

Formatted: English (United Kingdom)

9 Release Management Approach

9.1 Release Management Approach

The Release and Configuration Management approach articulated in this document is more stringent than would ordinarily be the case for the Integration Test and User Integration Test phases of a programme. It is with good reason that a more stringent approach has been articulated.

The MHHS Programme has multiple Programme Participants that will be involved in the various stages of testing, far more so than in most other programmes. In a programme with no other, or a small number of other participants the impact of a sub-optimal release will be less than with a large number of additional participants. If the MHHS Programme attempts a release into the test environments that fails or has issues then it is not only the MHHS Programme that will be impacted, it will be the other external industry parties that are impacted. These impacts could result in amongst others; delays, re-work, erosion of stakeholder & partner confidence.

The MHHS Programme has a responsibility to ensure that a robust release process is utilised in the testing phases of the programme in order to minimise any adverse impact on Programme Participants. A successful, error free release does not occur by accident, it is as a result of utilising defined processes and a robust plan. Fail to Plan – Plan to Fail!

9.2 Release Content

A Release can contain a combination of some, or all of;

- Base functionality, i.e. the initial Major Release
- Change Requests
- Defect Fixes

The MHHS Change Request Process needs to ensure that the Impact Assessment process considers the Release Management Approach required to implement the CR.

The IA response to a CR should determine which parties are impacted by any Change. For a Release by any of the Core Providers (DCC, Recco, DIP, Electralink, Helix / Elexon, St. Clements) the principle is that all MHHS Programme Participants are impacted as all MHHS Programme Participants involved in testing will be connecting to the Core Providers.

Formatted: Indent: Left: 0 cm

~~9.3 Release Rehearsals~~

~~Serious consideration should be given to rehearsing the release process prior to the initial release of code into the SIT environment prior to the commencement of the CIT phase.~~

10 Release Classification & Frequency

10.1 Release Types

- **Major** - release of software that contains significant additions of functionality.
- **Minor** - release of software that contains minor additions of functionality.
- **Patch** - release of software that bundles defect fixes, for example a scheduled weekly release of defect fixes.
- **Emergency** – release of software which contains a fix for a blocking testing defect that can not wait until the next scheduled Patch Release.

The MHHS releases will cover defect fixes which are identified in SIT and/or Qualification testing. Releases could also include changes which have been through the MHHS change governance framework.

10.2 Release Frequency

The Release frequency for Major and Minor Releases has yet to be defined within the Programme and planning will be reflected in future versions of this document as the MHHS Programme Plan matures. The Release frequency for Patch Releases and Emergency Releases will be ad hoc, although it is anticipated that Patch Releases will be more frequent than Emergency Releases.

Formatted: Caption, Keep with next

Release Type	**Definition**	**Frequency**	**Notice Required**	**RFC Required in ADO**
Major	Release of software that contains significant additions of functionality			Yes
Minor	Release of software that contains minor additions of functionality	Weekly	1 Business Day	Yes
Patch	release of software that bundles defect fixes, for example a scheduled weekly release of defect fixes.	Weekly (Every Monday morning)	1 Business Day	Yes
Emergency	release of software which contains a fix for a blocking testing defect that can not wait until the next scheduled Patch Release	Ad hoc	1 hour	Yes

Table 3: Release Type Details

11 Naming Convention

MHHS releases will be named with 4 separate digits, as below in Figure ~~4~~5, which increment depending on the content of the release. Each digit represents the version increment for each of the four types of release, as described in section 10.1, Release Types. Each Central Systems Provider will utilise the MHHS Programme Naming Convention for their releases. As each Central Systems Provider can implement code releases independent of each it is not anticipated that each Central Systems Provider will have the same release number, other than following the initial release.

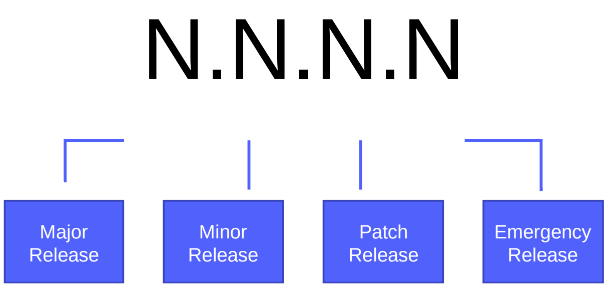


Figure 5 Release Naming Convention

11.1 Naming Versioning

Component	Meaning	Usage
1.n.n.n	Major Release	Release of software that contains significant additions of functionality.
n.1.n.n	Minor Release	Release of software that contains minor additions of functionality.
n.n.1.n	Patch Release	Release of software that bundles defect fixes, for example a scheduled weekly release of defect fixes.
n.n.n.1	Emergency Release	Release of software which contains a fix for a blocking testing defect that can not wait until the next scheduled Patch Release.

Table 4 Naming Versioning

- Versioning following the first major release will be 1.0.0.0
- Versioning following the first minor release will be 1.1.0.0
- If the first major release is followed by an emergency release the versioning would be 1.0.0.1
- Each release type will reset the subordinate version numbers. So, if after the first major release there has zero minor releases, one patch release and seven emergency releases the version number would be 1.0.1.7. If there is another, second patch release the version number will be 1.0.2.0
- Each major release will reset the subordinate version numbers. So, following the second major release the version number would be 2.0.0.0

12 Release Lifecycle

The Release Lifecycle has 5 stages as described in Figure ~~5~~6 below, each stage has a number of steps. The stages describe specific release management and other relevant activities in each step. The steps for each stage are in section 12.4, Deployment and 12.5, Review & Closure.

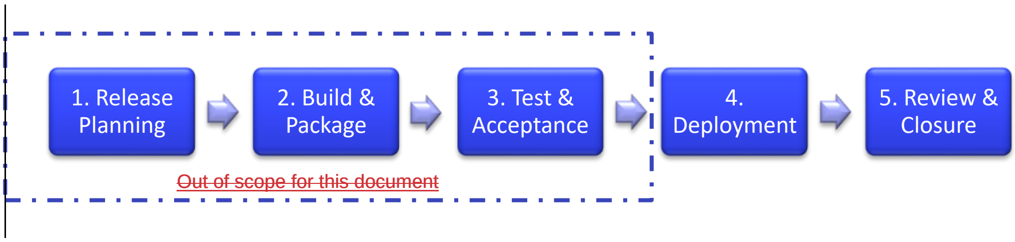


Figure 6 Release Lifecycle

12.1 Release Planning

Out of scope for this document, refer to document [MHHS-DEL 763 Release Management Procedure v0.2] for details of how MHHS manages Release Planning.

12.2 Build & Package

Out of scope for this document, all Core Systems Providers will have their own processes for how they manage the Build process.

12.3 Test & Acceptance

Out of scope for this document, all Core Systems Providers will have their own processes for how they manage their PIT Testing phase prior to the deployment of the initial Release. The MHHS Programme will undertake PIT Assurance as a Quality Gate for CIT / SIT. Each Programme Party is responsible for resolving SIT and UIT defects in accordance with the MHHS Defect Management Approach.

12.4 Deployment

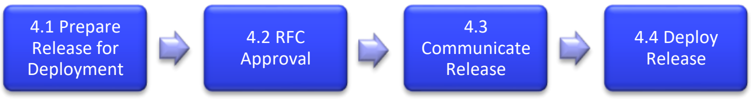


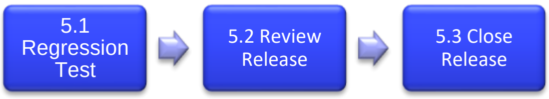
Figure 7 Deployment Lifecycle

Step	Description of Activities	Inputs and Outputs
4.1 Prepare Release for Deployment	Prepare for deployment to Test Environments, create implementation/rollout plan including resources required, timings, agreed downtime, stakeholders impacted and approvals / communications required, update runbook.	**Input**: GNG meeting approval / Appropriate Programme governance milestone met, Defects, known issues / Workarounds, Release package, installation instructions, updated runbook, Release summary, Release Note, **Output**: Implementation/rollout plan. RFC
4.2 RFC Approval	Raise RFC (Request for Change) for deployment into Test environments~~ estate and attend ~~[. The Release ~~Control Board] as required. ~~Manager reviews the request. NB – The RFC Process applies to both CRs and Defect fixes.	**Input**: MHHS Programme Change Management process **Output**: Approved RFC
4.3 Communicate Release	Create & send out required communications. Notify Defect Triage Teams of known issues, workarounds and resolutions to previously identified issues / incidents and existing workarounds.	**Input**: Release Communications Plan, communications materials, Implementation/Rollout Plan. **Output**: Communications to stakeholders.
4.4 Deploy Release	Deploy release within agreed RFC window~~ and in line with installation instructions~~. Log issues / resolutions found when deploying into test environments~~, and any errors in installation instructions~~. Receive sign-off that everything is working as expected, and RFC can be closed.	**Input**: ~~Release package,~~ Implementation/Rollout Plan, ~~Installation instructions, ~~Approved RFC. **Output**: Release ~~package ~~deployed successfully / change rolled back. Report on deployment and issues encountered / fixed in deployment and associated confirmation/smoke testing. Summary of release status communicated to all impacted stakeholders.

Table 5 Release Deployment Process

Formatted: Normal, Space After: 8 pt, Line spacing: Multiple 1.08 li

12.5 Review and Closure



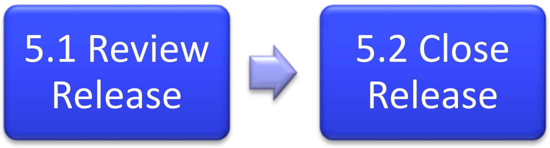


Figure 8 Review and Closure Lifecycle

Step	Description of Activities	Inputs and Outputs
~~5.1 Regression Test~~	~~Following deployment a Regression Test will be executed to ensure that the build has deployed successfully.~~	~~**Inputs:** Deployment Report~~ ~~**Outputs:** Regression Test Report~~
5.~~2~~1 Review Release	~~Post-Implementation Review questionnaire to be sent out by Release Management and filled in by involved stakeholders (deployment teams, DIP Provider) in order to confirm that the deployment met its objectives and if issues were encountered to capture and feed into 'Lessons Learned'.~~ Where appropriate a lessons learnt exercise will be undertaken from all relevant stakeholders, Central Parties and PPs If successful continue to 5.~~3~~ ~~Release Closure~~ ~~Rollback: If Defect return to~~ 2~~.~~ ~~Build & Package, otherwise~~	~~**Inputs**: RFC, Implementation/Rollout Plan, Deployment Report, Regression Test Report~~ **Inputs**: RFC **Outputs**: Success / Fix forward / Rollback ~~Sign-off from~~In case of defect fix, the defect is marked as 'Ready for Retest' and the relevant stakeholders ~~that~~ are made aware the release ~~met its objectives and~~ is complete so that the retest can ~~be closed~~take place. Lessons Learnt if rollout was unsuccessful or there were any deviations from the plan.

	~~return to 4. Deployment~~ Close Release	
5.~~3~~2 Close Release	Close RFC~~.~~, the RFC can be closed when the release is deployed and not when each defect has been successfully retested. Agree any work-off actions.	**Inputs**: Successful completion of release or rollback activities **Outputs**: Work-off actions. Release closure communicated to stakeholders

Table 6 Review and Closure Process

12.6 Detailed Descriptions of Major Inputs and Outputs

The major inputs into the Release Management process are:

Inputs	Description	Input is expected from / at what Process Step
~~GNG meeting approval / Appropriate Programme governance milestone met~~GNG discussion with Test team (Test Manager and Test Co-ordinators)	For the Major Releases, aligned to Programme Milestones it is likely that there will be Programme Governance Meetings that approve the Entry Gate(s) into various testing phases. Eg, decisions regarding uplift to IR7.3 or continue with IR5.5 ~~Code deployment and Test Data Load will be phased for CIT participants based on their respective entry Interval for CIT.~~ For Minor Releases, Patch Releases and Emergency Releases Programme Governance approval will not be required.	Programme Governance Meeting / 4.1 (FTIG)
Approved RFC for deployment	RFC for release deployment into the Test environments reviewed and approved by the ~~[release governance forum, name TBD]~~ SI Release Manager within MHHS Programme Release Management process for SIT, and the Qualification test Release Manager for UIT.	MHHS Programme Release Management / Step 4.1

Table 7 Major Release Management Inputs

The major outputs of the Release Management process are:

Outputs	Description	Outputs delivered to / at what Process Step
Release Notes	~~A document provided by the appropriate Central Party development team containing information related to the prepared release.~~The RFC will contain any relevant release note.	All impacted stakeholders in accordance with the agreed Release Note distribution list / Step 4.1
RFC document	RFC document for deployments to be submitted for approval by the ~~appropriate governing forum~~relevant Central Party.	MHHS Release Management / Step 4.1
Release Communication Plan	~~Document describing what communications are required for the Release. The plan formally defines who is given what information, when that information must be delivered and what communication channels will be used to deliver the information.~~Communication to MS cohort teams channels informing Participants of timing and content for releases.	Project Stakeholders / Step 4.3

Communications Materials	Specific instructions or briefing notes required to notify and inform stakeholders about a Release.	All impacted stakeholders in accordance with the agreed Release Plan / Step 4.4, Step 5.2

Table 8 Major Release Management Outputs

12.7 Process Quality Control Points

A control point implemented into a Release Management Process is one procedure or process output included into the chain of a process which can “block” the following implementation of a process if the result of the work performed isn’t produced properly and doesn’t meet the requirements.

The list of control points for the Release Management Process can be found below in Table 9.

#	Control Points	Process Step No.	Responsible
1	RFC Raised	4.1	Release Manager Governance: SI Release ~~Control Board~~Manager
2	Programme Governance Approval (for Major Releases only)	4.1	Release Manager Governance: Programme Governance Forum(s)
3	Environments are built / maintained at required configuration baseline.	4.1	Environments Manager ~~Governance: Release Control Board~~
4	Release Package and installation instructions are approved (if applicable)	4.1	Build Manager ~~Governance: Release Control Board~~
5	List of known issues, workarounds and fixes to existing issues/workarounds is approved by / agreed with Resolving Defect Teams and distributed as appropriate.	4.1	Release Manager ~~Governance: Release Control Board~~
6	Release plans produced and approved by the ~~[~~Release ~~Control Board]~~Manager	4.1	Release Manager ~~Governance: Release Control Board~~
7	RFC Approval	4.2	Release Manager ~~Governance: Release Control Board~~
8	Issues and resolutions are captured at time of deployment, and any amendments required in installation instructions are noted	4.4	Release Manager ~~Governance: Release Control Board~~
9	Rollout completion / closure is approved	5.2	Release Manager ~~Governance: Release Control Board~~
10	Rollout report is delivered & Lessons Learnt are captured	5.2	Release Manager ~~Governance: Release Control Board~~

Table 9 Release Management Process Control Points

12.8 Emergency Release

An Emergency Release is a type of release which is a quick fix to an emergency problem, e.g. a blocking defect at a critical stage of testing.

Emergency releases should be related to critical or blocking defects and should not be a result of bad planning or miscommunications within the normal release process.

To quickly implement the Emergency Release, the steps of the process are the minimum necessary to deploy quickly and with quality. The request for an emergency release is likely to be initiated by the test phase manager that is impacted by a blocking defect.

Figure ~~8~~9 below shows the Emergency Release Workflow:

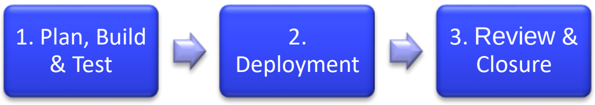


Figure 9 Emergency Release Lifecycle

NB: All emergency patches/hotfixes must be included into the next available release/patch, so that it can be tested and brought into a regular release, removing the risk of overwriting an emergency patch/hotfix.

Step	Description of Activities	Inputs and Outputs
1. Plan, Build & Test	Review and replicate defect, identify impacted components and agree target resolution. Identify configuration baseline for impacted component(s) and data. Implement change to CIs, unit & component testing, build release package. Create release note. Plan testing required. Deploy release package Test Environments, complete identified testing required.	**Input**: Defect identified ~~**Output**: Release package, installation instructions, Release summary, Release Note~~ **Output**: RFC Evidence of testing completed in Development environments. Key Roles: Release Manager, Development Manager, Dev Ops, Test Manager, Defect Manager, Deployment Team
2. Deployment	Prepare for deployment to test environment, create implementation/rollout plan including resources required, timings, agreed downtime (if any), stakeholders impacted and approvals / communications required. Raise RFC for deployment into test environment. Deploy release within agreed RFC window and in line with installation instructions. Log issues / resolutions found when deploying into test environment, and any errors in installation instructions.	~~**Input**: Release package, installation instructions, Release summary, Release Note~~ **Input**: RFC Release communications required **Output**: Release~~ package~~ deployed successfully / change rolled back Report on deployment and issues encountered / fixed in deployment and associated confirmation/smoke testing. Summary of release status communicated to all impacted stakeholders.
3. Review & Close	Update relevant documentation. ~~Merge changes to CIs back into all other in-flight releases~~ Conduct release retrospective with all stakeholders (If applicable). Agree work-off actions~~.~~. Confirm release closure with stakeholders.	**Input**: ~~Release deployment report~~Updated RFC **Output**: Work-off actions, ~~updates to continuous improvement log, ~~update to release schedule and communication of release completion

Table 10 Emergency Release Process

12.9 Release Build

A build should be a set of files and/or configuration items, placed in a meaningful directory structure that can be installed following a documented installation procedure, preferably mostly automated. A source control system needs to be in place which will allow for incremental builds and the ability at any time to regenerate any particular previous build. All code must be checked into and deployed from the source control system to each test environment.

Once a build is prepared for a Release, it must be made static and only updated as a result of a Request for Change (RFC). Builds may not be “fixed” or “tweaked” in place once they enter the Release Management Process without an RFC and a new identifiable build being created.

12.10 Release Notes

The Release Note ~~that~~ is ~~delivered along with~~part of the ~~build describes all versions, statuses, defect statuses, RFCs as well as instructions on how to construct, operate and back out the build.~~ RFC. The Release Note should ideally include;

- Programme or project name;
- Title of the release including reason for release (e.g. Build x.y - release for CIT Testing in SIT or Migration Testing);
- Date of release & release number (e.g. 16-Aug-2023, 1.0.0.0);
- ~~Name and title of party build is released to;~~
- ~~Name and title of authorised party release is issued by & details of other authorised signatories;~~
- ~~Tag used in Source Control System (to allow the regeneration of that particular build):~~
- ~~If applicable, any user manuals & guides (documentation itself, version number, date):~~
- ~~Database schema, if applicable (schematic diagrams, description, version number, date).~~
- ~~Target hardware environment: if applicable (description, diagram, version, configuration elements).~~
- ~~Target operating system: if applicable (description, version, configuration elements).~~
- ~~Bill of Materials (BOM) (for each file and configuration item – the version number, date, filename);~~
- ~~Release content list (a plain English description for each of the above files and configuration items);~~
- List of defects fixed in the release (ID, severity, summary);
- ~~List of outstanding defects not fixed in the release (ID, severity, summary, work around);~~
- List of Change Requests addressed in the release (ID, title, summary);
- ~~Installation Instructions - broken down into 3 sections as below. Each step within each section should have step number, step description, who will perform the step, estimate of how long it will take to execute. Helpful is the inclusion of a summary of the total time for each step:~~
    - ~~Release Pre-Requisites - e.g. installation accounts have relevant admin rights, any software required to deploy the release already installed on environment, any backups required for backout have been taken, system inventory list to be cross checked with the code package delivered;~~
    - ~~Release Instructions - step by step and relevant to the release environment. These should be at the lowest level to avoid any doubt/misunderstanding. It should include steps to stop/start servers, any configuration items that need to be set up, what locations files should be copied to, what permissions they should have, what commands need to be executed, any changes that need to be made to databases - *everything* that needs to happen to make the release work on the new environment. It should include checks/expected results so that the installer knows whether it is working or not. It should also include any baseline data requirements:~~
        - ~~Often there is a Build Tool associated with a Build. If so then the version number, configuration items etc need to be specified in the Release Instructions.~~
- ~~Post Release Checks – items that can be used as a quick sanity check that this particular build has been a success.~~
- ~~Backout Plan - detailed instructions on how to rollback to the previous release. If this is different depending on where the installer has reached within the installation instructions then the rollback plan needs to cover all eventualities (e.g. there might be a set of instructions for backout if installation fails~~

~~before Step 5, and a different set of instructions if it fails after step 5). Each step within each set of instructions should have step number, step description, who will perform the step, estimate of how long it will take to execute. Helpful would be a summary of total time to backout for each set of instructions.~~

- ~~Some builds can be applied without disruption to the availability of the service, others may require a database to be frozen for the duration of the build, others may require the server to be taken down for the duration of a build. These conditions should be highlighted as a separate item in the Release Note.~~

~~Additionally, Release Notes:~~

- ~~For Major releases, are to be submitted at least 5 working days in advance of the Release Control Board (RCB), for all other releases Release Notes are to be submitted 1 day in advance of the Release Control Board, and~~
- ~~Must reflect the true picture of the relevant test environment and therefore must include all application and product versions at that point of time.~~
- All Central Parties except St Clements and Helix will include a Rollback Plan with the RFC. St Clements and Helix will do fix forward as rollback would introduce complexity and risk for their integrated components.

12.11 Release Communications and Stakeholder Engagement

As part of the overall management of releases, the MHHS Release Manager, with input from the Central Parties Release Managers, as appropriate, ~~-~~will ensure that the correct communications are sent to all impacted stakeholders to ensure effective, clear and consistent understanding of the release, impacts and progress is maintained.

In the case of a Major Release, the Release Manager may create a specific Release Communications Plan for a release detailing the stakeholders, types, purpose and frequency of communications.~~ The Communications Plan may include the following;~~

- ~~Summary of the agreed Release Plan~~
- ~~Release Summary, Release Notes & Installation instructions~~
- ~~Release Status at each process step~~

13 Release Governance

~~13.1 Release Control Board~~

~~The Release Control Board will govern each release at a detailed level.~~ All releases will require a Request for Change to be raised by the resolver party and approved~~.~~

~~The Release Control Board is designed to bring together all stakeholders necessary to ensure a release moves through~~ by the ~~planning, packaging~~SI Release Manager for SIT and ~~deployment phases in a controlled manner. Meeting required gate criteria and managing changes through~~ the ~~process.~~

~~Major releases, and certainly~~ Qualification Test Release Manager for UIT. The Release Managers will ensure the ~~initial major releases~~design team, test team, all impacted PPs and the Central parties are ~~likely to be more tightly governed than later minor, patch and emergency releases. The first major release will be the most tightly governed as it will be the first and there will be no experience to draw from. Given~~aware of the upcoming release and ready for the ~~criticality of ensuring that the programme milestones are achieved there may be a requirement for a release rehearsal, either physical or paper~~next steps before approval.

~~Release Control Board~~	
~~**Purpose:**~~	~~Brings together all required stakeholders to provide a formal control point for each release, tracking and agreeing progress through the lifecycle.~~ ~~Governs the Release~~
~~**Responsibilities:**~~	~~Formal agreement to move a release for movement through release lifecycle.~~ ~~Retains a record of the decisions and versions of releases as they are promoted through the environments.~~ ~~Identify changes to agreed release baseline and approve or reject these changes.~~ ~~Provides a Go / No-Go decision for the Release to be deployed.~~
~~**Frequency:**~~	~~Daily/Weekly – as required~~
~~**Attendees:**~~	~~Attendees will vary depending on the release type. More senior representation may be required for major releases. As the programme progresses and patch / emergency releases become the norm the attendees will flex in order to ensure that there is sufficient functional representation whilst ensuring that the meeting doesn't become a resource drain.~~ • ~~Release Manager (Chair, Control and administer meetings, scheduling and prioritising agenda items, recording approvals and distributing outcomes),~~ • ~~Environment Manager, Development Team Lead, Test Manager, Defect Manager~~

~~Table 10 Release Control Board~~

~~13.2 Meeting Guidelines~~

~~Meeting Name~~	~~Members~~	~~Content / Purpose~~	~~Frequency~~
~~Release Kick-Off~~ ~~(Initial Release Planning)~~	~~Release Manager, Environments Manager, Test Manager, Defects Manager, Development Team Lead~~	~~Determine the requirements for the release.~~ ~~Likely to be a more formal meeting for major releases, and ad-hoc potentially not required for minor and patch releases. Required for Emergency Releases.~~	~~Once~~
~~Release Plan Review~~	~~Release Manager, Environments Manager, Development Team Lead(s), others as required~~	~~Review the release plans, ensuring that all of the activities that are required for the release deployment have appropriate steps at an appropriate level of granularity.~~ ~~Ensure all teams have the same understanding of the plan.~~ ~~Identify any issues or risks.~~	~~As required~~
~~Go / No-Go~~	~~Release Manager,~~ ~~Environments Manager, Development Team Lead, Test Manager~~ ~~Others, tbc~~	~~Assesses the readiness of the release, determines whether all of the required steps have been completed.~~ ~~Makes a Go / No-Go decision for release deployment.~~	~~Once~~

~~*Table 11 Meeting Guidelines*~~

~~13.2.1 Meeting Output~~

~~All meeting output will be distributed to all meeting attendees. In addition Core System Provider and Test Participant representatives will receive the meeting output.~~

14 Roles & Responsibilities

The key roles which appear in this approach document are:

Role	Definition
SI Environments & Release Manager	Overall accountability for the operational efficiency and effectiveness of environments and release management. • Coordination for planning and maintaining schedule for test environment usage. • Documenting the environment provision plan with the SI Test Team when available. • Raising and/or coordinating any environment Defect in ADO and liaising with relevant environment teams. • Assuring environments are stable during the Test window in conjunction with the relevant environment managers. • Managing & Tracking change in general across all environments. • Involved in Defect Triage meetings on a regular basis. Assuring that all required systems are connected & working as expected in the test environment prior to test execution.
SI Release Manager	A key role within the Release & Deployment Process, whose purpose is to manage the overall release process, to act as a point of escalation for process participants. • Responsible for release management plan. • Checking Release Notes when patch is delivered to determine which Defects can be set to retest. • Coordinating with the SI Defect & environment managers when required.
DevOps Teams	A key role in release package development, whose purpose is to build the release package, provide release content information and manage Release Note content within whole Release Lifecycle.
SI Environment Manager	Implements environment management strategy, policies and practices to provide and maintain the required non-production environments to support the programme.
Deployment Team	Performs the deployment of the Release into the test environments.
SI Test Manager	Responsible for managing one or more test phases.
SI Defect Manager	• Leadership & communication of Defect management process. • Point of Escalation for defect issues and defect SLAs. • Regular Defect Status Reporting. • Running Defect Triage Panel and managing the audience. • Analysis of Defects to assist in project decision-making activities. • Liaising upstream with senior stakeholders and downstream with Test Participants Test and Programme teams.

Role	Definition
	• Single point of contact for any user level access management towards Defect Management Tool (ADO). Involved in Defect status calls / Triage meetings.
SI Triage Team	• Representation from SI Design, Test and Programme teams. • Review and analyse the newly raised Defects (along with Severity and Priority of the Defect) by different Test Participants' Test team and drive relevant actions. • Change the assigned Priority and Severity if required as a part of triage assessment. • Involved in regular Defect Triage Meetings. Involved in changing the status from 'New' to 'Open' if a newly raised Defect is accepted by SI Triage Team.
Test Defect Manager (per Test Participant)	• Reviewing & managing the quality of the Defect Submitted by Tester (along with Severity and Priority of the Defect). • Involved in Defect Triage meetings on a regular basis. • Point of contact for the SI Defect Manager and Test Execution Team. • Driving Retest of Defects that have been delivered into the test environment. Reviewing, accepting, and closing Defects that have been successfully retested in the test environment.
QT Test Lead	Will be responsible for Managing Qualification Test Phase
QT Defect Manager	• Leadership & communication of Defect management process. • Point of Escalation for defect issues and defect SLAs. • Regular Defect Status Reporting. • Running Defect Triage Panel and managing the audience. • Analysis of Defects to assist in project decision-making activities. • Liaising upstream with senior stakeholders and downstream with Test Participants Test and Programme teams. • Single point of contact for any user level access management towards Defect Management Tool (ADO). Involved in Defect status calls / Triage meetings.
QT Triage Team	• Representation from SI Design, Test and Programme teams. • Review and analyse the newly raised Defects (along with Severity and Priority of the Defect) by different Test Participants' Test team and drive relevant actions. • Change the assigned Priority and Severity if required as a part of triage assessment. • Involved in regular Defect Triage Meetings. • Involved in changing the status from 'New' to 'Open' if a newly raised Defect is accepted by SI Triage Team.
Tester (Test Participant)	• Submitting new Defects. • Defect retest.

Role	Definition
	- Involved in Defect Triage meetings on a regular basis wherever necessary. Retesting fixed Defects as per the release notes/info in Defect Management Tool (ADO).
SI Design Authority	Provides Design concurrence for the design activities under the MHHS solution for environments.
SRO ~~TMAG~~SITAG Chair	- Point of escalation for System Integrator. Oversight of Environment Management process.
BSC and/or RECCO	- Point of escalation for System Integrator during Qualification.
PMO	- Set up and highlight agenda for monthly EWG meetings. - General communication and escalation point. Administering documentation.
Security Points of Contact	- Responsible as central points of contact for obtaining certificates and any other information related certificates Roles are defined in [REF-06] MHHS ~~DIP 094~~ DEL1197 - Interface Code of Connection ~~Guide include DIP Service Providers~~
~~DIP Representatives~~	~~- Responsible for certificate registration~~ ~~Roles are defined in [REF-06] MHHS DIP 094 Interface Code of Connection Guide include Senior Responsible Officer (SRO), Appointed Responsible Officer (ARO) and Technical Contact (TC)~~

Table 11 Release & Configuration Management Roles

14.1 RACI Matrix

This section defines which Role is responsible, accountable, informed or consulted during key process activities.

The roles below follow the RACI model: R – Responsible, A – Accountable, C – Consult, I – Inform:

Activity	Release Manager	Development Manager	Dev Ops	Environment Manager	Test Manager	Defect Manager	Core System Providers	Test Participants
4. Deployment								
4.1 Prepare Release for Deployment								
~~Create & agree Rollout Plans & Deployment Schedule~~RFC Review	R, A	C	C	C	C	C	C	I
4.2 RFC Approval								
RFC Approval	R, A	C	C	C	C	C	I	I
~~Deployment Go / No-Go meeting~~	~~R, A~~	~~C~~	~~C~~	~~C~~	~~C~~	~~C~~	~~R, C~~	~~I~~
4.3 Communicate Release								
~~Send out Communication Plan & Materials~~Issue Release Communications	R, A	I	I	I	I	~~I~~ I	C, I	I
4.4 Deploy Release								
Execute Implementation / Rollout Plan	R, A	I	I	I	I	I	R	I
Perform Remediation if deployment failed	A	C	C	C	C	I	R	I
Verify that all aspects of the Release are as expected	A	I	I	I	I	I	R	I
Create report / log of issues & resolutions during deployment	A	I	I	I	I	I	R	I
Conduct Rollback if any issues have been found after Rollout	A	C	I	I	I	C	R	I
5. Review and Closure								
5.1 Review Release								
~~Conduct PIR~~Confirm Release Deployment	R, A	C	C	C	C	C	R	I
~~Confirm Rollout results and Release Closure~~	~~R, A~~	~~I~~	~~I~~	~~I~~	~~I~~	~~I~~	~~R~~	~~I~~
~~Document Lessons Learned~~	~~R, A~~	~~I~~	~~I~~	~~I~~	~~I~~	~~C~~	~~R~~	~~I~~
5.2 Close Release								
Close RFC	R, A	C	C	C	C	C	C, I	I

Table 12 RACI Matrix for Release Management Activities

~~15 Toolsets~~

~~In the context of this document for the creation and deployment of any release the teams involved use a number of tools to drive the process.~~

~~Tool~~	~~Purpose~~
~~Azure DevOps~~	~~Service for support teams to plan work, collaborate on code development, and build and deploy applications.~~ ~~Used for Defect Management: defects registration and handling.~~ ~~Used for version-controlled repository for release candidates, components and software configuration items (CI's).~~
~~ServiceNow~~	~~The Service Now Change Man~~[illegible] ~~module is used to record and manage changes as part of the Release Man~~[illegible]~~ocess.~~
~~AnOther~~	~~Automated system for ma~~[illegible]~~ng, maintenance, software delivery and environment provisioning tool.~~ ~~Versions and variants of the platform image and install media are held in the xxx version control system [GitLab]. The version control system also holds the installation packages for the DIP software.~~
~~GitLab~~	~~Source code repository and configuration control~~

EXAMPLES

~~Table 14 Release and Build Toolsets~~

~~16 Configuration Management~~

~~At this stage it is not possible to provide detail on the subject headings below. When further information is available within the Programme the document will be updated.~~

- ~~Enduring Design Hub~~
- ~~Interface Code of Connection~~
- ~~Configuration – connection details for TPs to connect their own systems to central systems.~~
- ~~Backups – detailing process and frequency of backing up the environment including copies of data and configs.~~
- ~~Exports - detailing process and frequency of backing up the environment including copies of data and configs.~~
- ~~Refreshes - detailing process and plan for data refreshes.~~
- ~~Rollbacks – documenting the process to roll back the environment to a previous version.~~
- ~~Business data loading – documenting the process to load business data.~~

~~16.1 Business Data Loading~~

~~The process for defining which data is loaded and how the data is loaded is under development.~~

~~16.1.1 Process~~

~~TBD~~

~~16.1.2 Mechanism~~

~~TBD~~

~~16.1.3 Timings / Frequency~~

~~TBD~~

~~16.2 Exports~~

~~The process for exporting data should be an FYI to participants, unless an outage is required in which case it is anticipated that the Release Process defined in this document is utilised.~~

~~16.2.1 Process~~

~~TBD~~

~~16.2.2 Mechanism~~

~~TBD~~

~~16.2.3 Timings / Frequency~~

~~TBD – Likely to be ad hoc.~~

~~16.3 Backups~~

~~The process for Backups should be an FYI to participants, unless an outage is required in which case it is anticipated that the Release Process defined in this document is utilised.~~

~~16.3.1 Process~~

~~TBD~~

~~16.3.2 Mechanism~~

~~TBD~~

~~16.3.3 Timings / Frequency~~

~~TBD~~

~~16.4 Refreshes~~

~~The process for Refreshes has yet to be determined, however it is anticipated that Refreshes will follow the Release Process defined in this document.~~

~~16.4.1 Process~~

~~TBD~~

~~16.4.2 Mechanism~~

~~TBD~~

~~16.4.3 Timings / Frequency~~

~~TBD~~

~~16.5 Rollbacks~~

~~The process for Rollbacks has yet to be determined, however it is anticipated that Refreshes will follow the Release Process defined in this document.~~

~~16.5.1 Process~~

~~TBD~~

~~16.5.2 Mechanism~~

~~TBD~~

~~16.5.3 Timings / Frequency~~

~~TBD~~

Formatted: Caption, Space After: 0 pt, Line spacing: single