Release Management Procedure

Document name	Process_Design_Document_RM_V0.1.docx
Version	0.1
Status	☒ In progress ☐ Approved ☐ Validated
Update date	2025/XX/XX
Owner	
Level of confidentiality	☐Public ☒Internal ☐Limited ☐Confidential

Document Control

CHANGE HISTORY:

AUTHOR(s)	VERSION	DATE	CHANGES
Fadia BOUAZZA	V0.1	2025/05/02	Document creation

APPROVAL

ROLE	NAME	DATE
Foundation Office Lead	Li-Kang KUANG	2025/07/07

Table of Contents

1. Introduction

1.1. Main definition

A release is a rollout of new or modified IT service from the development to the production environments. It may include software, hardware, documentation, processes, or any combination.

A **Minor Release** includes small enhancements, updates, or fixes that are backward-compatible and pose minimal risk. More frequent and has a lighter process (no SIP Assessment, Runbook update instead of creation, etc.)
While a **Major Release** introduces brand new services, major architectural changes, or impacts multiple systems and domains. Needs to follow the full process, including extensive UATs, SIP assessment, and CAB approval.

A **release package** is a packaged collection of one or more releases that are tested and deployed together into the live environment.

Example:
Major Release: Deployment of Syensqo's brand new ITSM platform
Minor Release: Adding patches or minor configuration changes
Release Package: Adding 2 new modules at the same time (MIM + Release Management)

1.2. Objectives, challenges & benefits

The Release Management process consists of planning, scheduling and controlling the deployment process or the "release" of IT services. Its goal is to ensure that new or changed services are deployed smoothly and effectively, minimizing risks and ensuring quality.

The implementation of Release Management process allows to:

- Have a secure go-live with less incidents and rollbacks
- Ensure the service being released is mature enough, since all necessary actions realized and verified through the Checklist
- Get a complete and thorough Knowledge Transfer, with the Runbook definition, Knowledge Base structure, Trainings, User guides, etc.
- Provide easy and efficient hypercare since the operational run teams are involved early on the process

1.3. Actors

There are 5 main roles in the Release Management process are:

1. Delivery Lead

The Delivery Lead is accountable for the overall design, performance, and continual improvement of the Release Management process. He is usually a senior manager with the ability and authority to ensure the process is rolled out and used by all stakeholders.

They are responsible for:

- Defining and maintaining the Release Management process framework, roles, and policies
- Ensuring alignment with overall ITSM strategy and governance standards
- Monitoring process performance and ensure it delivers expected value
- Identifying areas of improvement and lead process optimization initiatives
- Validating major process evolutions, KPIs, and governance models
- Supporting the Release Manager and ensuring coordination with other ITSM practices
- Representing the process in audits and executive-level discussions

ITIL role: Release Process Owner

2. Delivery Manager

The Delivery Manager is responsible for day-to-day facilitation of the global change process within his scope. He ensures the quality and consistency of the process and acts as the local focal point for Release Management, reviewing all releases in their scope, validating the planning, and following through the go/no-go decisions.

They are responsible for:

- Governing the end-to-end execution of the Release Management process on their scope
- Leading and facilitating release steering committees and decision boards

- Managing the overall release planning
- Monitoring KPIs and driving continuous improvement
- Maintaining and updating process rules and documentation
- Ensuring internal alignment and assess domain-specific impacts
- Liaising with technical teams and requesters

ITIL role: Release Manager

3. Service Owner/Product Owner

The Service Owner or Product Owner identifies the need for a release and initiates the process. He is responsible for formally submitting the release and providing the necessary context and inputs.

They are responsible for:

- Identifying and justifying the need for a release (triggers, objectives, etc.)
- Creating and submitting the release ticket in the ITSM tool
- Providing key information and documentation (Move2Run Checklist & Runbook)
- Responding to questions during the review process
- Preparing the communication before go-live
- Monitoring the release post go-live, providing support and collecting feedback

ITIL role: Release Requester

4. Technical team

The Technical team is responsible for the technical assessment, planning, implementation and testing of the release.

They are responsible for:

- Contributing to technical analysis

- Defining implementation, testing, and rollback plans
- Estimating effort, risks, and technical constraints
- Building and configuring the release packages
- Conducting tests and UATs
- Conducting post-implementation validations
- Providing input for the lessons learnt when relevant

5. Portfolio Board

The Portfolio Board acts as a bridge between project governance, financial control, and release execution, to ensure the financial viability and formal authorization of major IT releases by approving the project budget and initiating the release process upon validation.

2. Process description

2.1. Process triggers, inputs, outputs

1. **Triggers -** *What initiates the process ?*

Release process triggers are events or conditions that initiate a release process. These triggers signal that it's time to plan, build, test, or deploy a new or changed service component.

For Minor Releases the most common triggers are the following:

- Completion of a Development Cycle: When a sprint ends with ready-to-deploy features (small enhancements)
- Change Management Process: A formally approved change via the CAB initiates a release
- Security Patch or Compliance requirement: Security or compliance-driven deadlines may necessitate a release
- Scheduled Maintenance: Predefined maintenance patches can drive regular or batch deployments
- Infrastructure or Service Update: Upgrades, middleware changes, or cloud migrations require a coordinated release
- Evolution Requests: Business or technical improvement requests that aim to incrementally enhance existing services or applications.

Examples:

- *A new feature approved in the product roadmap*
- *A patch or fix to address a security vulnerability*
- *An infrastructure change (e.g., cloud migration)*

Major Releases are usually delivered as part of Formal Projects, which can have the following triggers:

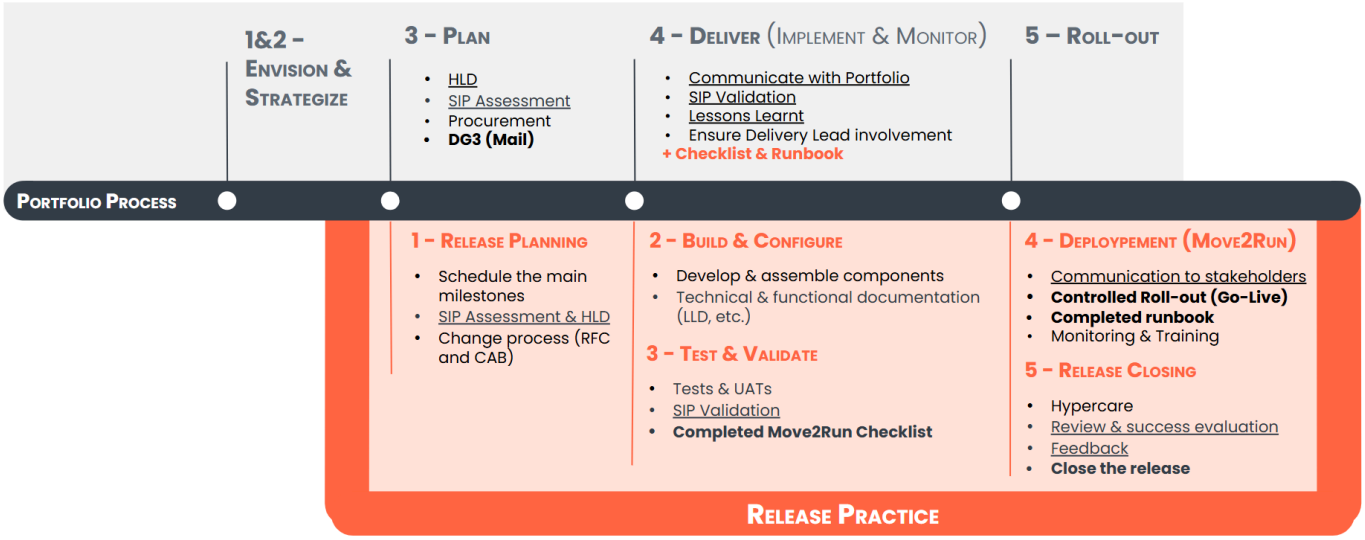
- Business Request: Initiated by business stakeholders to deliver new capabilities, improve user experience, or respond to market opportunities
- Demand Management: Stemming from portfolio or capacity planning activities where

business and IT align on priorities and allocate resources for large-scale changes

- End of Support for Legacy Systems: Decommissioning old services leads to migration releases

Formal Projects are followed through and approved by the Portfolio Process, which has 5 main steps: Envision, Strategize, Plan, Deliver and Roll-out.
The **Release Management Process gets triggered once the Project is validated** on by the Portfolio Board and moves to the Plan Phase. **It takes over to further deep dive the Porftolio's deliver & roll-out phases** with full fledged activities, that will be described in the next section.

The illustration below visually represents the close interconnection alignment of the 2 processes, with some common activities and gates.



2. Input - *What formally enters the process ?*

A **Release ticket** is created and submitted in the ITSM tool (BMC Helix), containing all necessary information for planning, scheduling and controlling.

Typical input elements:

- Description of the release
- Impact and risk analysis

- Desired deployment window
- Change Requests (RFCs)

3. **Outputs -** *What does the process deliver ?*

A **controlled and successful delivery of new or changed services** into the live environment, ensuring smooth Move2Run transition and minimizing risks

Examples of outputs:

- Communication and coordination with stakeholders
- Lessons learned and post release review
- Completed Runbook and Move2Run Checklist
- Updated CMDB and service documentation
- Backout Plans and Rollbacks

2.2. Process activities

The Release Management process includes 5 main activities :

1. **Release Planning**
    - Define the scope and objectives of the release
    - Identify required resources
    - Assess risks and dependencies (SIP Assessment)
    - Establish a release schedule
    - RFC Submission, Change Assessment & CAB Approval

2. **Build & Configuration**
    - Develop and assemble the release components
    - Assemble the CIs
    - Implement SIP recommendations
    - Create release packages

- Initiate technical and functional documentation (Service Description, HLD, LLD, Escalation Matrix, etc.)

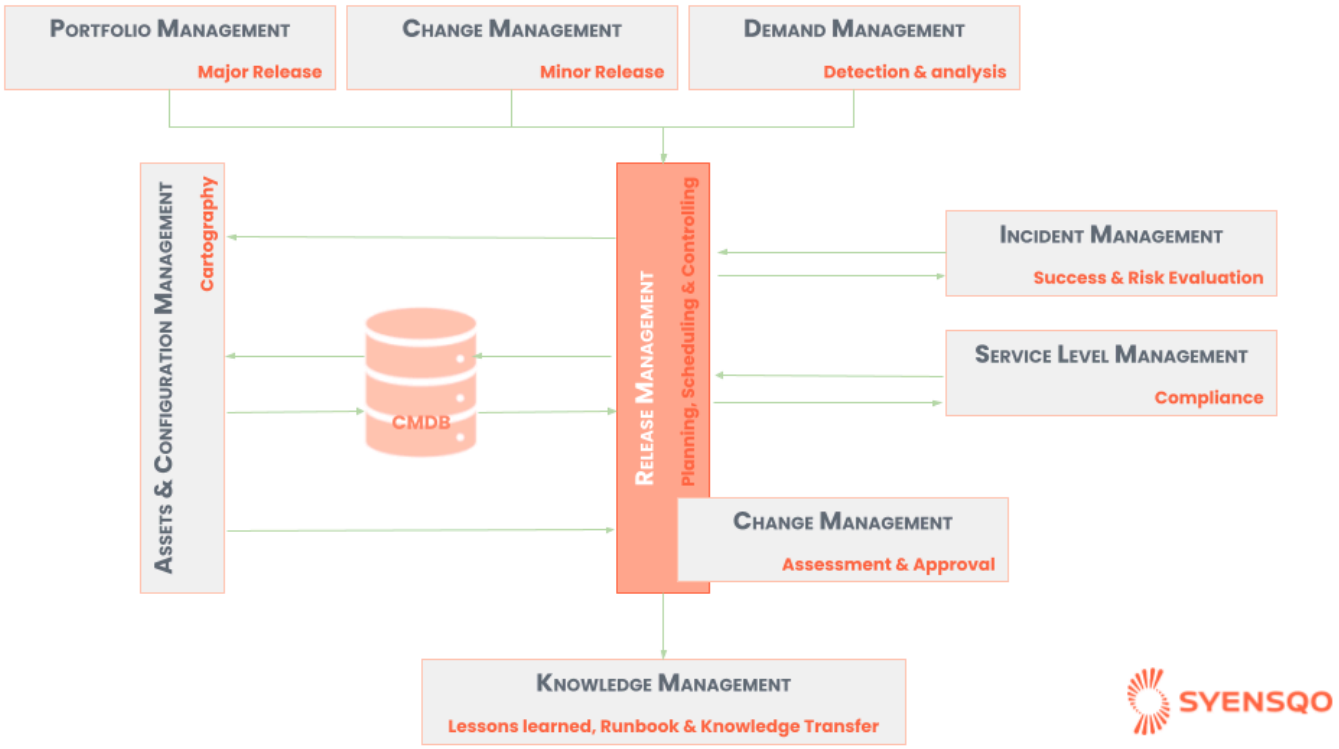
3. **Test & Validation**
   - Conduct functional and non-functional testing
   - Perform user acceptance testing (UAT)
   - Validate security and compliance requirements (SIP Validation)
   - Ensure all the items in the Move2Run Checklist are completed

4. **Deployment (Move2Run)**
   - Communication to all stakeholders and end users
   - Ensure a well-coordinated and controlled roll-out
   - Monitor for issues during deployment
   - Finalize Runbook and provide training for end-users and support teams

5. **Release Closing**
   - Hypercare
   - Conduct a release review to evaluate success
   - Identify lessons learned for future improvements
   - Gather feedback from users and stakeholders
   - Close the release and update records in the Configuration Management Database (CMDB)

2.3. Link with other ITSM processes

The Release Management process relies on structured interactions with several other practices to ensure effective control, risk mitigation, and alignment with business priorities.

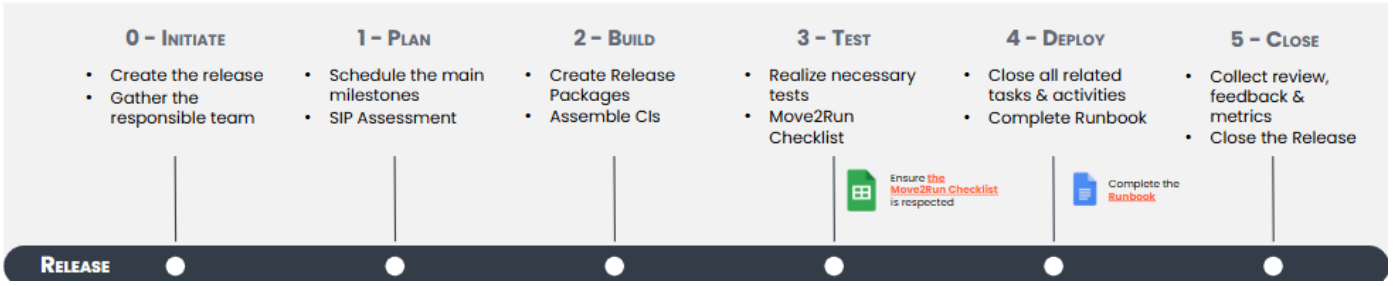


- **Demand Management :** identifies and anticipates demands that later translate into releases planned and prioritized based on the information collected
- **Portfolio Management :** Projects and programs are some of the main triggers for Major Releases. Portfolio Management is therefore a key input provider, triggering new releases aligned with strategic initiatives. The two are closely intertwined as shown in the earlier illustration (*cf. Process Triggers*)
- **Change Management :** A release consists of one or more changes that need to be reviewed and approved in the CAB before deployment. The Change Requestor needs to submit an RFC during the Plan Phase of the release and follow the Change until the Closure phase (*cf. details in the workflow description*)
- **Incident Management :** The incident support organization needs to be defined before the go-live of a service, as the escalation matrix of an incident is included in the Runbook. Additionally, incident records post deployment help define the success of a release and are closely followed to trigger a rollback if necessary

- **Knowledge Management :** The Knowledge Transfer is a key step in the Move2Run process. All the documents produced during the release process (Runbook, Checklist, etc.) need to be stored and made available in the Knowledge Base;
- **Assets & Configuration Management :** helps tracking what's in the release via Configuration Items (CIs), while the release management ensures the CMDB is updated before go live
- **Service Level Management :** SLAs for a new service need to be defined before the go-live (included in the Runbook). While releases must also comply to the pre-defined SLAs and contractual agreements.

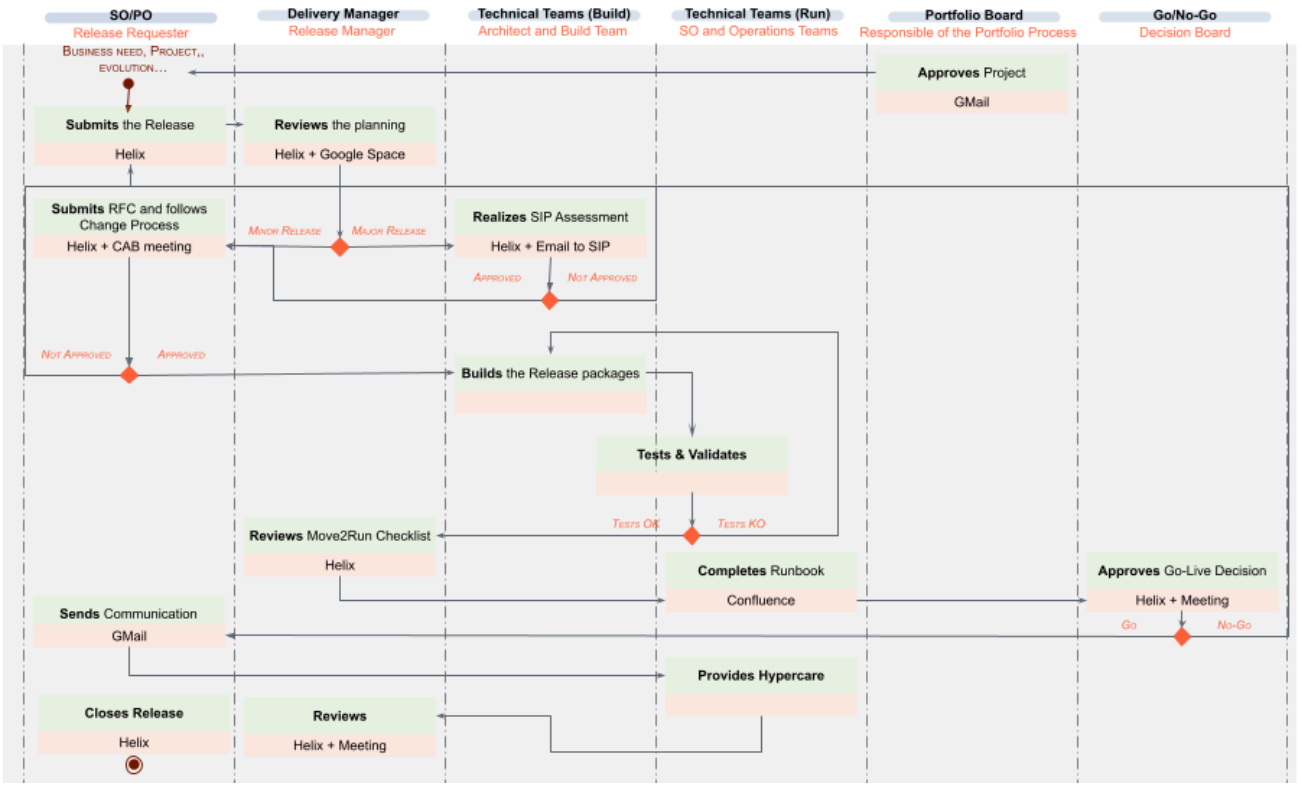
3. Process Workflow

As described in the Process activities, the Release Management Practice has five main steps, which you can see below along with the main gates and validation documents associated with it.



3.1. Flowchart

You can find below the detailed flowchart of the process activities



0 – Initiation

The process is triggered by a business need, an IT project, or a required system evolution (*cf. full list of triggers*). This early step ensures that all stakeholders are aligned before any technical activity begins.

- The **Release Requester (SO)** submits the release request through **Helix**
- The **Portfolio Board** validates and approves the overall project via **email**, confirming its alignment with strategic objectives, in case of a Major Release

1 – Planning

Once the project is approved, detailed planning begins. This stage covers change control, release scoping, and operational readiness checks.

- The **Delivery Manager** reviews the release timeline and milestones using **Helix** and **Google Space**
- The Release Requester submits a **Request for Change (RFC)** and follows the **Change Advisory Board (CAB)** process
    - If the RFC is **not approved**, the process stops or requires rework
    - If the RFC is **approved**, the release planning moves forward
- For major releases, the **Technical Build Team** performs **SIP assessment**
    - SIP Tool submitted by email to the SIP Team, and updates followed through Helix
    - If **approved**, the release moves to the build phase; if **not approved**, adjustments are needed

2 – Build

Once planning is finalized, the technical team constructs all components required for the release.

- The **Build Team** develops the release packages based on functional and architectural requirements
- Packages are assembled in preparation for testing and deployment
- Technical documentation is initiated

3 – Test

This step ensures the quality and stability of the release before production deployment.

- The **Technical Teams** execute **testing and validation** activities.
    - If **tests pass**, the release proceeds
    - If **tests fail**, corrective actions are applied and testing is repeated

4 – Move2Run

This transitional phase ensures operational readiness for Go-Live and handover to Run teams.

- The **Delivery Manager** reviews the **Move2Run checklist**, confirming all prerequisites are met
- The **Runbook** is completed by the technical teams in **Confluence**, documenting key steps, contact points, and rollback procedures
- A **Go/No-Go meeting** is held to validate readiness
    - Approval is granted through **Helix** following or during the meeting
    - If **Go**, the deployment is authorized
    - If **No-Go**, the deployment is delayed or canceled
- A **communication plan** is sent via **GMail** to inform users and stakeholders of the release timeline and scope

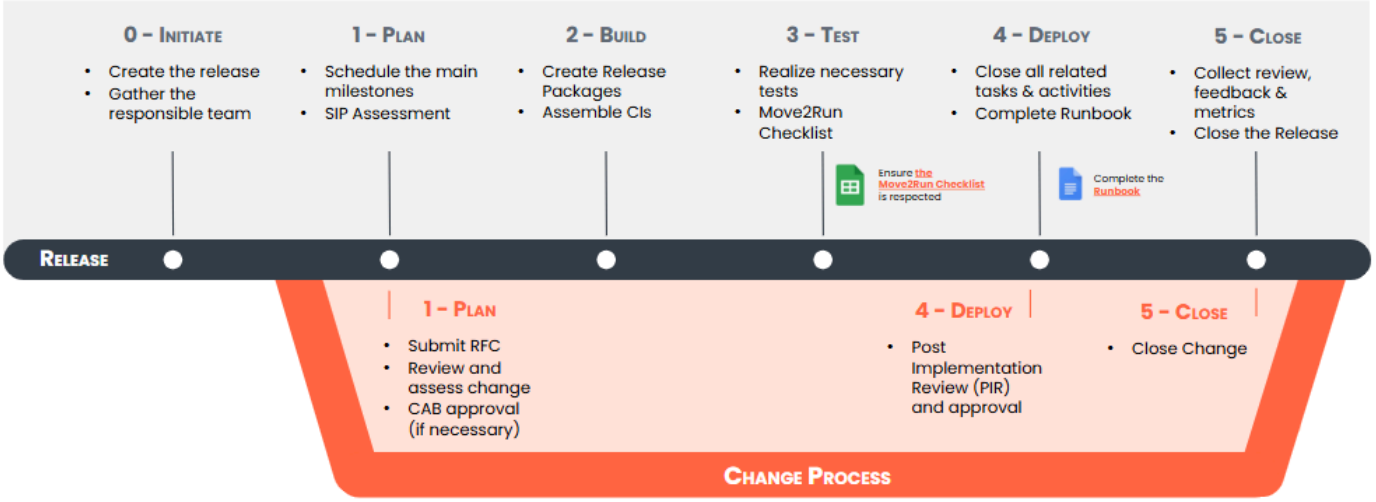
5 – Hypercare & Closure

The final phase focuses on live deployment, immediate support, and formal closure of the release cycle.

- The **Run Team** provides **hypercare** support to monitor stability handle incidents, and trigger rollback plan if needed
- A **post-release review** is conducted to assess outcomes and gather lessons learned for continuous improvement
- Once stable, the **Release Requester** closes the release in **Helix**

3.2. Release Management & Change Management Process Interconnection

The Release Practice is closely interconnected with the Change Management Process, as we can see in the illustration below.



1. **As part of release planning**, related Requests for Change **(RFCs) are submitted**. These RFCs must be assessed for impact, risks, and dependencies. If the change is high risk or significant, it may require **Change Advisory Board (CAB) approval**.
2. **Once the Change is approved**, the Release process moves to the **Build & Configuration** of the changed components.
3. The Release package then **undergoes thorough testing** for functional and technical validation
4. but also **assessment of the implemented change**. The **Post-Implementation Review (PIR)** in the change process confirms the change achieved its objectives without adverse impact.
5. Deployment marks the culmination of both processes. The release is deployed to the live environment, while The **Post-Implementation Review (PIR)** in the change process confirms the change achieved its objectives without adverse impact.
6. The **release process closing phase** contains the collection of feedback, review of metrics, and documentation of lessons learned, while the **corresponding change is also closed upon successful deployment** and verification.

Release Management provides the **structure and coordination** to deliver changes to production, while **Change Management** ensures that each change within a release is **properly assessed, approved, and documented**.

4. Roles & responsibilities

A Responsible (R), Accountable (A), Consulted (C), and Informed (I) (RACI) matrix is used to describe the roles and responsibilities of the stakeholders involved in the implementation of the Release management process.

The Release Management process is led by the roles presented on chapter 1.3:

4.1. Waterfall Methodology

		SO (Release Requester)	DM (Release Manager)	Technical Teams (Build)	Technical Teams (Run)	Portfolio Board	IT Leads
1.1	*Initiate the Release*	R,A	C	-	-	-	-
1.2	*Plan the main milestones*	R	A	-	-	I	-
1.3	*SIP Assessment*	A	I	R	C	I	-
-	*Create RFC and Follow-up with the Change Ticket review and approval*	R, A	I	C	C	-	-
2.1	*Build & Configure the release packages*	A	I	R	C	-	-
3.1	*Realize tests*	A	I	R	R	-	-
3.2	*Ensure the Move2Run Checklist is completed*	R	A	C	C	I	–
4.1	*Complete the Runbook*	A	I	C	R	-	-
4.2	*Monitor & Follow-up the Runbook*	R	A	C	C	I	-
4.3	*Determine Go/No Go decision for Go Live*	C	A	I	I	C	R
4.4	*Send communication to stakeholders and end users*	R,A	I	C	C	I	I

5.1	*Provide Hypercare and monitoring*	A	I	R	R	-	-
5.2	*Close the Release*	R,A	C	I	I	-	-

Roles & Responsibilities – Release Management - Waterfall

4.2. Agile Methodology

In Agile methodology, there is a high emphasis and expectation on team ownership and accountability, with the traceability of the decisions being made through the team agreement. So the RACI is slightly modified to take into account this specificity, and the (*) Roles can be taken by a member of the Agile team depending on the sprint repartition.

		Product Manager	Product Owner	Scrum Master	Team	Release Requester (*)	Release Manager (*)	DevOps (*)
1.1	*Initiate the Release*	C	C	C	A/R	R,A	C	R
1.2	*Plan the main milestones*	C	C	C	R	R	A	R
1.3	*SIP Assessment*	A	C	C	R	R,A	I	R
-	*Create RFC and Follow-up with the Change Ticket review and approval*	I	I	I	A/R	A	I	R
2.1	*Build & Configure the release packages*	-	-	-	A/R	A	I	R
3.1	*Realize tests*	-	-	-	A/R	A	I	R
3.2	*Ensure the Move2Run Checklist is completed*	-	-	-	A/R	R	A	R
4.1	*Complete the Runbook*	-	-	-	A/R	A	I	R
4.2	*Monitor & Follow-up the Runbook*	-	-	-	-	R	A	-
4.3	*Determine Go/No Go decision for Go Live*	A	C	C	C/R	C	A	C
4.4	*Send communication to stakeholders and end users*	A	C	-	R	R,A	I	-
5.1	*Provide Hypercare and*	-	-	-	A/R	A	I	-

	monitoring							
5.2	*Close the Release*	-	-	-	A/R	R,A	C	R

Roles & Responsibilities – Release Management - Agile

5. Metrics & KPIs

5.1. Process Success Factors and their KPIs

To monitor the effectiveness of the Release Management process and ensure it delivers both operational performance and user satisfaction, **4 process success factors (PSFs)** have been identified.

PSF 1 - Properly involving all relevant stakeholders early in the process

Engaging relevant stakeholders early ensures alignment on goals, requirements, budget and timelines. It improves risk identification, knowledge transfer, and increases the overall success and quality of the release. Below are some of the main actions to ensure this PSF is respected:

- Service Owner taking on the role of Release Requestor
- Clear communication throughout the process
- Alignment with the Portfolio board on the release planning and budget validation
- Implement change approval processes (CAB, e-CAB, etc.) where needed
- Provide complete knowledge transfer

PSF 2 – Following the standardized tools, process and templates

Using standardized tools, processes, and templates ensures consistency, reduces errors, accelerates execution, and enables smoother collaboration across teams involved in the release process.

- Using the defined templates and checklists for consistency (Move2Run Checklist, Runbook, etc.)
- Establishing clear timelines and release calendars
- Following the documented Release Management workflow in BMC Helix
- Adhering to versioning and naming conventions to ensure clarity and traceability
- Undergoing SIP Assessment

PSF 3 - Identifying and preventing release risks early

Proactively identifying and mitigating release risks early minimizes disruptions, enhances stability, and ensures smoother deployments. This can be ensured through:

- Preparing rollback plans preparation
- Ensuring backup systems are in place

- Track missing elements through the Move2Run Checklist
- Conduct thorough User Acceptance Testing

PSF 4 - Continuously Improving the Release Management Process

Ongoing improvement ensures the process adapts to evolving business and technological needs, increases maturity, and reduces unsuccessful releases, through

- Post-release reviews and success evaluation
- Feedback collection and lessons learned
- Document and share best practices and process updates across teams

5.2. Key Risks Indicators

To proactively manage and mitigate potential issues in the Release Management process, 4 key risk indicators (KRIs) have been identified.

KRI 1 - Stakeholder & Resource Risks (*cf. PSF1*)

- Lack of early involvement from key stakeholders
- Resource conflicts or unavailability during critical releases
- Unassigned ownership for key release activities
- Misalignment of the defined roles & responsibilities (*cf. earlier description*)

KRI 2 - Process & Tooling Risks (*cf. PSF2*)

- Unrealistic release timelines or frequent late release
- Multiple overlapping releases affecting the same domains
- Lack of adherence to standard release templates and processes

KRI 3 - Testing & Quality Risks (*cf. PSF3*)

- Insufficient test coverage or incomplete test scenarios
- Failed test cases not resolved before deployment
- Incomplete Move2Run Checklist high/medium priority tasks
- No rollback plan or untested rollback procedures

KRI 4 - Monitoring & Improvement Risks (*cf. PSF4*)

- No tracking of key release KPIs (*cf. metrics below*)

- Lack of visibility into release readiness
- Infrequent or missing post-release reviews

5.3. Key Performance Indicators & Metrics

To effectively monitor the health and maturity of the Release Management process, the following KPIs are grouped into three categories:

- **Frequency & Planning - to measure operational workload and agility**

KPI	Description	Calculation mode
Total Number of Releases	Measures overall release activity and workload per quarter/semester/year	Count of all releases (major + minor) per quarter/semester/year
% of Major Releases	Tracks the percentage of significant releases involving major changes	(Number of major releases / Total number of releases) × 100
% of Minor Releases	Tracks the percentage of smaller updates or enhancements in the release cycle	(Number of minor releases / Total number of releases) × 100
% of Overlapping Releases	Measures the percentage of releases that are overlapping, which could impact other releases	(Number of overlapping releases / Total number of releases) × 100

- **Quality & Reliability - to track risk, stability, and process control**

KPI	Description	Calculation mode
% of Releases Requiring a Rollbacks	Tracks the percentage of releases that had to be rolled back due to issues after deployment	(Number of releases requiring rollback / Total releases) * 100
% of Major Incidents Post Releases	Measures the percentage of major incidents caused by releases that significantly affect service or business	(Number of major incidents caused by releases / Total releases) * 100
% of Late Releases	Measures the percentage of releases that	(Number of late releases / Total

	missed their planned deployment date	planned releases) * 100
% of Successful Releases	Measures the percentage of releases that were deployed successfully: respecting the planning and all Move2Run criterias with no incidents or rollbacks	(Number of successful releases / Total releases) * 100

- **Continuous Improvement & Satisfaction - to ensure learning, follow-up, and stakeholder alignment**

KPI	Description	Calculation mode
Customer Satisfaction Score Regarding the Process	Measures stakeholders' satisfaction with the release management process	Average score from satisfaction form (out of 10)
Post-Release Review Coverage Rate	Tracks the percentage of releases that undergo post-release reviews to identify improvements	(Number of releases with post-release reviews / Total releases) × 100
Documented Lessons Learned and Best Practices	Measures how often lessons learned and best practices are documented after a release for future reference	(Number of releases with documented lessons / Total releases) × 100

6. Governance

The activities of the Release Management process are controlled through the following governance instances:

Instance	Purpose	Input (←) /Output (→)	Frequency	Participants
Release Steering Committee	Provide strategic oversight, resolve cross-team conflicts, and prioritize releases	← Release planning, status reports, KPIs → Strategic decisions, reprioritization, escalation resolutions	Monthly	Release Manager (DM), Release Requestors, Portfolio Board Manager
Go/No-Go Decision Board	Final go/no-go decision before production deployment	← Move2Run Checklist, Test summary, Rollback plan, Runbook → Go/No-Go decision, new deployment planning in case of delay	Before each release	Release Manager (DM), Release Requestor, Service Owner, IT Leads of the impacted domains
Post-Release Review	Identify lessons learned and improvement areas from the last release	← Release metrics, incidents, stakeholder feedback → Lessons learned, improvement actions, updated documentation	After each release	Release Manager (DM), Service Owner, Release Requestor, Operational Teams
Review of the Release Management Process	Evaluate the overall performance of the process, discuss KPIs and suggest improvements	← Quarterly KPIs & Lessons Learned → Action plan and roadmap for improvements	Quarterly	Release Practice Lead, Release Managers (DMs), Release Requestors

7. Tools & Deliverables

7.1. Runbook

Establishes a structured framework to support the run activities. Provides guidance to SOs, DMs, and operational teams in their daily operations and responsibilities.

- The V0 of the Runbook covers all critical elements that must be addressed immediately to secure Syensqo's successful takeover of run activities
- V0 completion before go live is mandatory
- V1 of the Runbook contains complementary optional elements that help provide guidance to (*coming soon*)
- The Runbook should be completed and updated regularly by the Service Owner to ensure accuracy of the content

7.2. Checklist

Verifies that **all necessary steps and activities have been completed and validated** before the go-live. Helps the SOs and DMs ensure everything is in order.

- All critical topics of the Checklist are included in the Runbook V0
- Can be leveraged to conduct additional checks and track missing elements that will need to be addressed post go-live
- Should be completed before go live for services released after September

7.3. BMC Helix

Ongoing Definition with SP13 Project

8. Glossary

Term	Definition
BMC Helix	ITSM platform. Central system of record for the release process
Release	Rollout of new or modified IT service from the development to the production environments
Minor Release	Includes small enhancements, updates, or fixes that are backward-compatible and pose minimal risk. More frequent and has a lighter process (no SIP Assessment, Runbook update instead of creation, etc.)
Major Release	Introduces brand new services, major architectural changes, or impacts multiple systems and domains. Often involves extensive UATs, SIP assessment, and CAB approval.
Release Package	Packaged collection of one or more releases that are tested and deployed together into the live environment
Release Management Process	Planning, scheduling and controlling the deployment process or the “release” of IT services. Its goal is to ensure that new or changed services are deployed smoothly and effectively, minimizing risks and ensuring quality.
CAB (Change Advisory Board)	Formal approval committee for Major Changes and Post Implementation Reviews. Chaired by the Change Process Manager
Move2Run Checklist	Essential tool that goes through all necessary steps and activities and ensure they have been completed and validated before the go-live
Runbook	Structured framework to support the run activities. Provides guidance to SOs, DMs, and operational teams in their daily operations and responsibilities.
Portfolio Board	Approves the budget of the project and triggers the release process

9. Appendices

Detailed interconnections between practices :

Process	What is the interconnection?	How is it materialized?	Who is involved?
Change Management	A release consists of one or more changes that need to be reviewed and approved in the CAB before deployment	The Change Requestor needs to submit an RFC during the Plan Phase of the release and follow the Change until the Closure phase (details in the workflow description)	Release Manager, Release Requester, Change Requester, Operational Teams
Knowledge Management	The Knowledge Transfer is a key step in the Move2Run process	All the documents produced during the release process (Runbook, Checklist, etc.) need to be stored and made available in the Knowledge Base	Release Manager, Service Owner, Architect, Knowledge Manager, Operational Teams
Service Level Management	- SLAs for a new service need to be defined before the go-live *(included in the Runbook)* - Releases must also comply to the pre-defined SLAs and contractual agreements	- The SLAs are included in the Runbook that is verified as part of the release - If a release does not comply to its SLAs,	Release Manager, Service Owner, Service Level Manager
Incident Management	- The incident support organization needs to be defined before the go-live of a service - Incident records post deployment help define the success of a release	- The escalation matrix of an incident is included in the Runbook that is verified as part of the release - The incidents post release are closely followed and if significant can trigger a rollback	Release Manager, Service Owner, Incident Manager, Operational Teams
Asset & Configuration Management	- Tracks what's in the release via Configuration Items (CIs) - Ensures the CMDB is updated before go live	- The Move2Run Checklist & Runbook ensure that the CIs are properly updated and listed in LeanIX	Release Manager, Service Owner, SAC Manager
Service Request Management	The Service Requests of a new service need to be defined before the go-live *(included in the Runbook)*	List of service requests included in the Runbook that is verified as part of the release	Release Manager, Service Owner, Service Request Manager
Portfolio Management	- Portfolio Management identifies and anticipates demands that later translate into releases planned and prioritized based on the information collected	The processes are closely intertwined (please refer to the details in the process trigger)	Release Manager, Portfolio Manager, IT Leads