

# Release Management Procedures

eDelivery internal

Exported on 02/19/2019

## Table of Contents

|     |  |    |
|-----|--|----|
| 1   | Release Management Procedures .....        | 3  |
| 2   | Table of Contents .....                    | 4  |
| 3   | Introduction .....                         | 5  |
| 3.1 | Scope .....                                | 5  |
| 3.2 | Objective.....                             | 5  |
| 4   | Process overall description .....          | 6  |
| 4.1 | DOMIBUS: .....                             | 6  |
| 4.2 | SML .....                                  | 6  |
| 4.3 | SMP .....                                  | 7  |
| 5   | Security and Vulnerability Assessment..... | 9  |
| 6   | Roles and responsibilities.....            | 10 |
| 7   | Versioning strategy .....                  | 11 |
| 8   | Release files preparation checklists.....  | 12 |
| 8.1 | SW release package TO checklist: .....     | 12 |

# 1 Release Management Procedures

## 2 Table of Contents

- [Table of Contents](#) (see page 4)
- [Introduction](#) (see page 5)
  - [Scope](#) (see page 5)
  - [Objective](#) (see page 5)
- [Process overall description](#) (see page 6)
  - [DOMIBUS:](#) (see page 6)
  - [SML](#) (see page 6)
  - [SMP](#) (see page 7)
- [Security and Vulnerability Assessment](#) (see page 9)
- [Roles and responsibilities](#) (see page 10)
- [Versioning strategy](#) (see page 11)

## 3 Introduction

The objective of this document is to achieve:

- a **structured approach** to **deploying** software components providing guidelines and support;

This document defines the release management workflow, the use of release management mechanisms, and the needs of different release steps, according to each component specificity .

### 3.1 Scope

This release management procedure covers activities needed to group and manage individual enhancements, corrections, adjustments into logical and recognizable modules (releases) that needs to be planned, developed, tested and implemented in a coherent way.

### 3.2 Objective

The aim of release management is to **build, test and deliver capabilities** in such a way that the delivered service **meets the requirements previously defined, complying with the defined quality standards**. New releases may arise from a need to add, remove, update or fix a service component. The **objective** of release management is to ensure that the defined releases in the yearly project roadmap are properly prepared and executed. This involves the follow up of the development activities, testing and communication, besides the actual steps necessary to prepare and execute the release itself.

## 4 Process overall description

The release planning is defined on a yearly basis in accordance with the project road map.

The component releases are segmented in two release moments:

- **Release Candidate (RC):** it is a beta version with potential to be a final product, which is ready to release unless significant bugs emerge. In this stage of product stabilization, all product features have been designed, coded and tested through one or more beta cycles with no known showstopper-class bugs. There could still be source code changes to fix defects, changes to documentation and data files, and peripheral code for test cases or utilities. No changes to the code are allowed except for the critical issues. If any critical problems are found and fixed, a new release candidate is issued.
- **Final Release (FR):** This is the production ready release, that should meet the criteria previously defined for it. All RC release criteria must be met; all critical bugs must be CLOSED; all known issues that might have an impact on the system should be documented as Known Issues; all the pertinent documentation must be updated accordingly and when applicable the proper communication must be planned.

As stated before each component has certain specific requirements when it comes to the release process, please find below a process overview per eDelivery component:

### 4.1 DOMIBUS:

According to the project roadmap the release manager should schedule the preparatory release meeting (at least 2 weeks before each release).

In this meeting should be present:

- Members of the development team involved in the development of the component (Tech lead)
- Release manager
- Project Officer
- Relevant elements of the testing team
- Representative of the SO (Support Office)
- Representative of the SMO (Stakeholders Management Office)

During this meeting:

- If not yet defined, the development freeze and testing periods must be set.
- Identify the documentation that requires to be updated, and assign the responsible element for the update
- Define the communication strategy and assign a responsible element
- Set deadline for the release page creation on the CEF digital portal in collaboration with the SMO
- Set responsible and deadline for the preparation of the release notes.
- Set responsible for the execution of the release package preparation check-list (there must be prepared upfront the SW release checklist for each component)
- Set dates and responsible for the SO training (training to be provided by the TO)

### 4.2 SML

According to the project roadmap the release manager should schedule the preparatory release meeting (at least 2 weeks before each release).

In this meeting should be present:

- Members of the development team involved in the development of the component (Tech lead)

- Release manager
- Project Officer
- Relevant elements of the testing team
- Representative of the SO
- Representative of the SMO

The SML besides being made available as a SW component, it is also **provided as a service**, therefore it is necessary to comply with the SML SLA. Stating that the **user communities should be notified 10 business** days in advance of any scheduled unavailability of the service (something a deployment of a new version will imply) in Acceptance and Production environments. This is a **responsibility of the SO**.

For RC the SML is deployed in acceptance for testing purposes (SO and user communities) after an acceptance period of maximum one month, FR should be deployed also in acceptance for the conclusion of the testing activities. Upon the successful testing the SML is released on the CEF digital portal. Following the release the SO should communicate the deployment in prod to the community 10 days later.

During this meeting:

- If not yet defined, the development freeze and testing periods must be set.
- Define the date of the formal user communities notification.
- Identify the documentation that requires to be updated, and assign the responsible element for the update
- Define the communication strategy and assign a responsible element
- Set deadline for the release page creation on the CEF digital portal in collaboration with the SMO
- Set Prod or Acceptance deployment timeframes
- Set responsible and deadline for the preparation of the release notes.
- Set responsible for the execution of the release package preparation check-list (there must be prepared upfront the SW release checklist for each component)
- Set dates and responsible for the SO training (training to be provided by the TO)

### 4.3 SMP

According to the project roadmap the release manager should schedule the preparatory release meeting (at least 2 weeks before each release).

In this meeting should be present:

- Members of the development team involved in the development of the component (Tech lead)
- Release manager
- Project Officer
- Relevant elements of the testing team
- Representative of the SO
- Representative of the SMO

The SMP is a distributed component in the eDelivery architecture, however currently the collaboration terms with the onboarding eHealth project require that the component is also deployed in a Acceptance environment used by eHealth. Unlike the SML, there is no need for formal notifications, it is required to only inform the stakeholders that a new version as been deployed

During this meeting:

- If not yet defined, the development freeze and testing periods must be set.
- Identify the documentation that requires to be updated, and assign the responsible element for the update
- Define the communication strategy and assign a responsible element
- Set deadline for the release page creation on the CEF digital portal in collaboration with the SMO
- Set responsible and deadline for the preparation of the release notes.

- Set responsible for the execution of the release package preparation check-list (there must be prepared upfront the SW release checklist for each component)
- Set dates and responsible for the SO training (training to be provided by the TO)



## 5 Security and Vulnerability Assessment

All the component releases should be audited from the security point of view by DIGIT` s Security Assurance team.

Upon the publishing of a RC version of a component the corresponding tech lead should submit the review request to the Security Assurance team. The steps to formalize the request can be found in the link below:

<https://webgate.ec.europa.eu/fpfis/wikis/pages/viewpage.action?spaceKey=SecurityAssurance&title=Web+Application+Vulnerability+Assessment+Procedure>

## 6 Roles and responsibilities

The table below shows a RACI matrix summarising the roles and tasks of the parties involved in release management activities.

|   | <b>Project Officer</b> | <b>Project/Release Manager</b> | <b>Component Technical Leader</b> | <b>Test Manager</b> | <b>SMO</b> | <b>SO</b> |
|---|------------------------|--------------------------------|-----------------------------------|---------------------|------------|-----------|
| <b>Schedule and plan release</b>                        | A/R                    | R                              | C                                 | C                   | I          | I         |
| <b>Update release calendar</b>                          | A                      | R                              | I                                 | I                   | I          | I         |
| <b>Maintain deployment checklist</b>                    | --                     | --                             | A/R                               | --                  | --         | --        |
| <b>Schedule the release preparation meeting</b>         | I/C                    | A/R                            | I                                 | I                   | I          | I         |
| <b>Write / Update documentation (UC, SAD, ICD, etc)</b> | I                      | A                              | A/R                               | A/R                 | I          | R         |
| <b>Preparation of the executable build</b>              | --                     | --                             | A/R                               | --                  | --         | --        |
| <b>Deploy release when applicable</b>                   | I                      | I                              | A/R                               | --                  | --         | --        |
| <b>Update and publish release page</b>                  | I                      | I                              | I                                 | --                  | A/R        |           |
| <b>Prepare release notes</b>                            | I                      | I                              | A/R                               | C                   | I          | --        |
| <b>Communication</b>                                    | C                      | I                              | --                                | --                  | A/R        | --        |
| <b>News articles and Social media</b>                   | C                      | --                             | --                                | --                  | A/R        | C/I       |
| <b>Communication with user communities</b>              | C                      | I                              | --                                | --                  | --         | A/R       |

RACI matrix: **A**= Accountable **R**=Responsible **C**=Consulted **I**=Inform

## 7 Versioning strategy

The identification of each release will be composed of five alphanumeric blocks separated with dots (.) and space (.). These blocks are detailed below.


***Component [Major release].[Minor release].[Maintenance release]***

Where a:

| <b>Release Type</b>          | <b>Characteristics of release type</b>   | <b>Release management implications</b>   |
|------------------------------|--|--|
| Major release                | <ul style="list-style-type: none"> <li>• New features</li> <li>• Supersedes preceding minor upgrades, releases, and emergency fixes.</li> <li>• Possibility of non-backward compatible changes</li> <li>• High impact on user</li> </ul>                 | <ul style="list-style-type: none"> <li>• Longer activity phases in the release activity workflow</li> <li>• Defined in the project roadmap</li> <li>• Few major releases per year</li> <li>• Generic Communication</li> </ul>  |
| Minor release                | <ul style="list-style-type: none"> <li>• Smaller incremental improvements or fixes to application features</li> <li>• Maintenance, such as technology upgrades</li> <li>• Supersedes preceding emergency fixes</li> <li>• Low impact on users</li> </ul> | <ul style="list-style-type: none"> <li>• Shorter activity phases in the release activity workflow</li> <li>• Defined in the project roadmap</li> <li>• Minor releases are more frequent than major releases</li> <li>• Generic Communication</li> </ul>  |
| Maintenance release (hotfix) | <ul style="list-style-type: none"> <li>• Quick fixes to repair problem</li> <li>• Measures to address blocking or critical issues raised by stakeholders</li> <li>• Corrections to known errors to meet high priority requirement</li> </ul>             | <ul style="list-style-type: none"> <li>• No extensive release management planning</li> <li>• Shorter activity phases in the release activity workflow</li> <li>• No need for official generic communication, only to specific communities impacted by the corrections (when applicable)</li> </ul> |

## 8 Release files preparation checklists

### 8.1 SW release package TO checklist:

| Components                     | Checklists  |  |
|--------------------------------|---|--|
| Domibus                        |  <p data-bbox="395 869 884 902">2018-04-23_Dom...ase_steps.docx</p>  |  |
| SMP                            | <ol style="list-style-type: none"> <li data-bbox="379 1115 1347 1178">1. Update <a href="#">README.md</a><sup>1</sup> if needed.</li> <li data-bbox="379 1189 1347 1252">2. Remove "-SNAPSHOT" from mvn version (development branch).</li> <li data-bbox="379 1263 1347 1326">3. Merge development to master.</li> <li data-bbox="379 1337 1347 1400">4. Build master and deploy on nexus.</li> <li data-bbox="379 1411 1347 1473">5. Tag version on master branch.</li> <li data-bbox="379 1485 1347 1547">6. Add "-SNAPSHOT" and increase version in mvn (development branch).</li> <li data-bbox="379 1559 1347 1621">7. Schedule training for the support team</li> </ol> | <input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/><br><input type="checkbox"/> |
| eTx adapter replacement plugin |   |  |

<sup>1</sup> <http://README.md>

