

Annex E

to EMSA/OP/24/2015 Tender Specifications

ITIL Procedures related with service transition & Service Operation

Background

Following decisions taken by the EMSA Executive Director in 2012 (for applying a coordinated and harmonised ICT services approach ensuring efficient management of the existing resources and alignment of business and technological needs of the Agency), the ICT Steering Group of the Agency initiated the process of aligning the ICT operational procedures with relevant ISO Standards – particularly ISO 20000. In this respect, the ITIL framework was adopted.

As of today the procedures as per table has been adopted and are applicable to all ICT operational services of the Agency and project deliveries associated to them.

Table 1 EMSA procedures on Service Transition & Service Operation

Applicable procedure	Reference
Service Transition	
Change Evaluation Management	Appendix A
Release and Deployment Management	Appendix B
Service Verification, Validation and Testing	Appendix C
Service Asset and Configuration Management	Appendix D
Service Operation	
Event & Incident Management	Appendix E
Problem Management	Appendix F

For a detailed description of each procedure, EMSA contractors should refer to the relevant appendix of this Annex.

Bidders are invited to align their project delivery methodology with the procedures listed in the table. In this respect is noted that TeamForge will be used at EMSA to manage all the procedures related to Service Transition.

Annex E- Appendix A

Change Evaluation Management

EMSA Change Evaluation and Management (CEM) encompasses changes to base-lined service assets and configuration items across the whole service lifecycle that may have an impact on the level of performance of all business services, and information and communication technology assets described within the scope of the EMSA ICT Landscape document.

Therefore the expected changes are related to new maritime application versions, new commercial off-the-shelf hardware and software, application and commercial software removal, and error, failure, and fault solving through suitable patches, including documentation.

Standard changes are pre-authorised changes that do not impact the quality of service of any maritime application and have an accepted and established procedure for a specific requirement and belong to the purview of Unit A.3 of EMSA. This refers to activities performed by ICT Service Desk such as restoring printing services, e-mail services, EMSA user logging issues, etc. A list of standard changes will be properly updated and communicated.

All the changes that have impact to the scope, schedule and/or budget of a maritime application are approved by EMSA according to the change management process. No changes will be initiated until properly documented and approved by EMSA. If a change proposal is released as a consequence of a horizontal change to the organization, impacting several areas (for instance Staff, working procedures, training, IT resources, services, applications, facilities, etc.) and potentially involving various RFC (Request for Change), the Change Advisory Board (CAB) will carry out a comprehensive and specific assessment of the proposal. By doing this assessment, the CAB will ensure that all risks have been identified and considered, and will provide guidelines on how to proceed. If required a final analysis and decision could be done by the EMSA ICT SG (ICT Steering Group).

Upon receiving a request for services, the Contractor will review the change documented in the request, analyse the scope of the request, and agree with EMSA on the impact of performing the evaluation of the change request. If EMSA accepts the impact (cost and/or time slippage), the Contractor assigns the resources required to evaluate the impact and suggest possible solutions.

- Simple changes that do not affect the contractual obligations can be prioritised, assigned, and activated on the authority of the Contractor, following agreement with EMSA.
- Complex changes, changes with significant risk potential or those that affect the contractual obligations between the Contractor and EMSA must be thoroughly evaluated and authorised before any action is taken.

The Contractor will ensure that the requests are dealt with in a timely and cost-effective manner.

The complete text of the CEM could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.

Annex E - Appendix B

Release and Deployment Management

The EMSA RDM process materializes the Change Plan for the implementation of changes in the services, which is the main outcome of the Change Management (CM) process.

The RDM process defines a roadmap to deploy application releases, agreed by the Maritime Application Teams and ICT, in order to minimize interferences onto the regular operation of the services and providing repeatable mechanisms to be used to deploy future releases into the production environments.

The goal of RDM is the protection of the production environment and its services through the use of formal procedures and checks working closely with the Change Management (CM), Service Asset and Configuration Management (SACM), and Service Validation, Verification and Testing (SVVT) processes.

Release packages are planned and designed to deploy changes into the production environment (after being tested and validated in the Test and Pre-Production environments according to what is defined by the Service Validation, Verification and Testing procedure) in an effective and efficient way. Release packages are related with the CM process to monitor their traceability.

The main objectives are:

- Plan releases and deployments in line with requirements resulting from approved changes and with structured implementation guidelines.
- Build, install, test¹, and deploy effective release packages of one or more changes ensuring minimum disruption to the production environment.
- Make sure that the change and associated IT assets respond to service level agreements. Review preparation for the release to ensure maximum successful deployments and minimal impact on the business and services.
- Promote stakeholders satisfaction through sound practices.

The Releases are classified into Major, Service Pack and Emergency software fixes with the following characteristics:

- **Major software release upgrades**, normally containing large areas of new functionalities. A major upgrade or release usually supersedes all preceding Service Packs, Patch Releases or Hot Fixes.
- **Service Pack release upgrades**, normally containing enhancements and fixes, some of which may have already been issued as emergency fixes. A Service Pack upgrade usually supersedes all preceding Patch Releases or Hot Fixes.
- **Emergency software fixes**, normally containing the urgent corrections to a small number of known Problems that are impacting business or technical functionalities.

Service pack and emergency software fixes should only be issued for blocking problems, while non-blocking issues shall be planned and included in Major Releases and Service Packs that are scheduled in advance and are issued during agreed intervals.

Major releases and Service Packs shall be defined in the project plan. The frequency can be revised at the end of each defined period for the following defined period. Every change that

¹ Refer to Service Validation, Verification and Testing (SVVT) for Test definitions and procedures

does not fall into the frequency defined/planned should be treated as an emergency software fix and be handled accordingly.

When planning new releases, the following guidelines should be taken into consideration:

- Major release upgrades: the interval should be of, at least, 6 months between releases.
- Service Pack upgrades: the interval should be of, at least, 6 weeks between service packs.
- Emergency software fixes: will be addressed and analysed on a case by case basis, taking into consideration the severity of the problems corrected in the fix.

Emergency fixes, due to their critical nature, do not follow the standard times defined hereafter and are treated on a case-by-case scenario.

Depending on the environment, the following intervals should be respected as well:

- Testing environment:
 - no restrictions will be applied, but any changes to this environment must be discussed and agreed between the Business Unit and the ICT.
 - Any change of any component should be agreed in advance between ICT and Business Units.
- Pre-production environment:
 - a maximum of 1 release per week;
 - Maximum 1–2 updates of App's Pre-Prod Environment.
- Production environment:
 - a minimum of 6 weeks between deployments of new releases;
 - 1 scheduled downtime for 1 update of App's Prod Environment;
 - Minimum 2 week 'stability period' of Pre-Prod before changes can be put into Prod Environment;
 - All releases must be scheduled to be applied from Tuesday to Thursday, trying to avoid dates immediately before or after a holiday period.

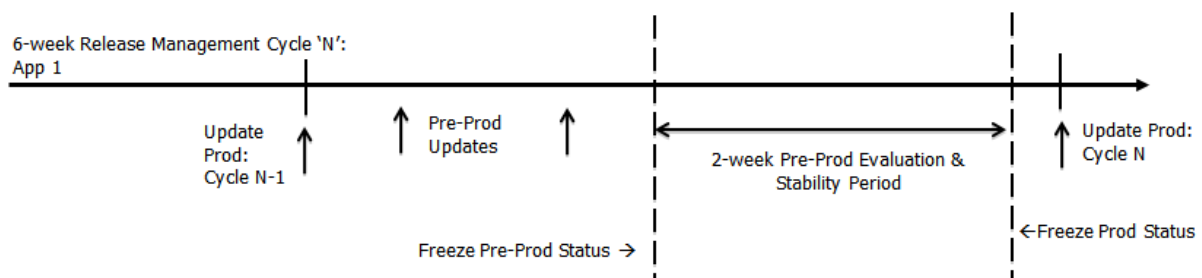


Figure 1 - 6 Weeks Lifecycle

6-week cycle of each Maritime Application is skewed by 1 week – to avoid updating all Prod environments at same time

Except if otherwise have been specifically agreed with the stakeholders of a maritime application, each version should be named in the following format: V.xx.yy.zz, where:

- xx – numeric sequential index representing a major release;
- yy – numeric sequential index representing a service pack;
- zz – numeric sequential index representing an emergency fix.

The software delivered by the contractor and all its components shall be built at EMSA using EMSA building environment before entering any EMSA environment.

Tools used in the build environment:

Licence Name	Version
Red Hat Enterprise Linux Server	6.4
Apache Maven	2.2.1 and 3.0.4
Apache Ant version	1.8.1
Apache Archiva	1.3.6
Hudson	3.0.0
SonarQube	4.0.0
Subversion	1.6
TeamForge	7.1

Requirements:

- Contractors must stick to and use these tools and provide deliverables using them. EMSA is open to discuss version upgrades if deemed necessary and justified,
- Build procedure and scripts must be fully integrated with the these tools,
- The software (including all related tools/components) must be built without human intervention,
- The release building process is documented and is a part of the release documentation.
- EMSA considers a new version delivered when:
- Source Code is successfully submitted to Subversion,
- All Maven external repositories are configured in Archiva,
- All needed components and libraries are downloaded from the external repositories, stored in and served by EMSA Archiva,
- Build procedures are configured in Hudson,
- Hudson build procedure is able to successfully build the system and all related tools/components.
- Results of the Quality Gate defined in the tender Annex on project delivery are evaluated and decision to accept or reject the release will be taken based on the conditions.
- The release building process is documented and is a part of the release documentation.

The complete text of the RDM could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.

Annex E - Appendix C

Service Verification, Validation and Testing

The aim of the SVVT process is to ensure that a new or changed service will provide the expected value, utility and warranty, to the business.

The benefits of testing properly services before being deployed into the production environments are basically the following:

- Customer's satisfaction increase, since the services are delivering the desired value to the users.
- Less requests to the Helpdesk for support, since services are operating as expected.
- Less incidents, since the functionalities and performance have been tested and match with the requirements.

In the context of this procedures software developers may be requested, in line with their contractual obligations to provide support in the development of the test Plan. The Annex 4 of the procedure shows examples of the content of a test plan (Template I) and test case specification (Template II).

The Test Plan constitutes the guideline by which the application will be tested, defining the acceptance criteria and the tests to be executed. In general the Test Plan makes reference to functional and non-functional tests under the following categories:

- Smoke tests: to check out if the main function of an application work properly, but not going into great details. They are put in place before the exhaustive testing activities or after the deployment on a different environment.
- Functional tests: to check out the functional behaviour of the application, including regression, in respect to specific function/feature.
- Non-Functional test: security, integration, load, stress, soak, and Business Continuity Facility (BCF) compliance. Load, stress and soak test descriptions must define all test scenarios to be executed, goals to be achieved, tools to be used and resources to be monitored.
- Acceptance (Business Validation) test: to validate that the business requirements are fulfilled by the new release. The 'Acceptance' formally acknowledges that the release has met the requirements once the release is deployed into the production environment. The 'Acceptance Criteria' should be defined by the business representatives and include at least a prioritized list of defined and measurable attributes, which are application-related, that must be satisfied to achieve the final acceptance.

The test procedures must aim at ensuring the compliance and quality of the release and to cause minimum disruption to the already in-place production environment. Furthermore should aim in verifying that the binaries resulted from the software built in-house are correct, can be used for installing the application in the production environment and once installed achieve the desired results.

All tests description must define all test scenarios to be executed, goals to be achieved, tools to be used, and resources to be monitored, when applicable¹.

Within the context of site acceptance activities a relevant step that is performed is Software Quality Assurance². Software Quality Assurance aims to:

- Assess and assure **Source Code Suitability** (Reliability, Correctness, Accuracy, Efficiency, and Usability) to identify issues/potential problems, propose solutions/changes and confirm implementation results.

¹ TestLink shall be used as platform to support the different testing activities, as in the case of the RDM procedure, the product(s) (i.e. Test Plan) shall be stored in TeamForge Version control directory and linked into the TeamForge Service Request

² Sonar tool will be used for software quality verification

- Assess and assure **Source Code Maintainability** (Understandability, Modifiability, Traceability, Testability, Portability, and Reusability) to identify issues/potential problems, propose solutions/changes and confirm implementation results.
- Gathering and interpret Source Code metrics.

In order to carry out this quality verification the following must be done:

- Perform Earlier Verifications:
 - focus on evaluating intermediate software builds and removing defects at coding time. SONAR is the tool selected for this activity. The Service Transition Team will inform in due time the Contractors on the criteria for testing and about the use of SONAR to perform this verification.
- Execute Test Plan:
 - focuses on executing tests cases/procedures approved.
- Perform Post-Product Verifications:
 - focus on evaluating final build quality or finding defect root causes after the product is complete.

Table 1 shows what tests have to be carried out in different environments although the MAT (Service Transition Team) is responsible for defining the profile of the testing activities considering the nature of the changes in the application and/or infrastructure.

Table 1. Environments vs. Test Types

TYPE OF TESTS	ENVIRONMENTS		
	TEST	PRE-PRODUCTION	PRODUCTION
SMOKE	Yes	Yes	Yes
FUNCTIONAL	Yes	Yes	
NON FUNCTIONAL		Yes	
ACCEPTANCE		Yes	

When applicable, the release will be deployed first in the Test environment in order to be functional tested.

If the results are successful then the release will be deployed in the Pre-Production environment to complete a second round of functional and technical testing. All deployments will be executed as described in the release documentation. Annex 5 shows an example of traceability matrix.

The test results are collected, analysed, and compared to the expected results in order to draft the Test Report (Annex 4, Template III). All test iterations and results will be included in Test Report and filled-in during the tests execution.

The complete text of the SVVT could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.

SVVT procedure, ANNEX 4

TEMPLATES**I. Test Plan (Standard IEEE 829-1998)****0. Unique Identification of the Test Plan****1. Introduction**

- 1.1 Summary of Features and Objects to be tested
- 1.2 References
- 1.3 Definitions and Acronyms

2. Tests Objects

- 2.1 Objects and respective test versions
- 2.2 Related documents
- 2.3. Bug Reports related to the objects to be tested
- 2.4 Objects no to be tested

3. Features to be tested (functional and non-functional)

- 3.1. All features and combinations to be tested
- 3.2 Related documents (Use and Test Cases with expected results)

4 Features not to be tested (functional and non-functional)

- 4.1 All features and combinations to not be tested
- 4.2 Rationale

5. Approach

- 5.1 General Testing Approach (type of tests)
- 5.2 Testing Approach for group of features/combination of features
- 5.3 Activities, techniques, and tools to be used

6. Limitations**7. Not successful Test Criteria****8. Suspension Test Criteria and requirements for re-starting**

- 8.1 Specification of criteria to suspend tests
- 8.2 Specification of activities to repeat tests

9. Test Tasks

- 9.1 Identification of tasks / procedures to prepare and execute tests
- 9.2 Identification of interconnections tasks

10. Test Environment

- 10.1 Specifications of the test environment
- 10.2 Specifications of the level of security
- 10.3. Specifications of special testing tools
- 10.4 Other needs required

11. Roles and Responsibilities (developers, testers, business team, A.3)**12. Knowledge and Training Requirements****13. Schedule****14. Risk Management****15. Approval**

II. Test Cases Specification (Standard IEEE 829-1998)

0. Unique Identification of the Test Case

1. Test Objects

- 1.1 Specification and conditions of the features to be tested

2 Input Specifications

- 2.1 Data Identification
- 2.2 Ordering
- 2.3. Values
- 2.3 States
- 2.4 Timing

3 Output Specifications

- 3.1 Data Identification
- 3.2 Ordering
- 3.3. Values
- 3.3 States
- 3.4 Timing

4. Special Procedures

5. Dependencies

III. Test Report

1. Executive Summary

- 1.1. Guidelines
- 1.2. Glossary

2. Testing Tools

3. Objectives

4. Test Environment

- 4.1. Description
- 4.2. Diagrams
- 4.3. Servers

5. Scripts

6. Test Execution

- 6.1. Execution Process
- 6.2. Sequence of Scenarios
- 6.3. Tasks between executions

7. Result Analysis

8. Conclusions and Recommendations

Annex E - Appendix D

Service Asset and Configuration Management

The Service Asset and Configuration Management (SACM) process of EMSA establishes a framework for the management of assets supporting the business services and also supports other service management processes such as Release and Deployment.

The SACM aims to maintain information about **Configuration Items (CI)** required to deliver an IT service, including their relationships. Maintain the information means, to keep updated records of the current CIs status, their change history and foreseen changes.

A **CI** is an IT asset or a combination of IT assets that may depend and have relationships with other IT processes.

CIs and relationships are stored in a **Configuration Management Database (CMDB)**. Definition of the processes to update CMDB and keeping it updated are the ultimate objectives of the SACM procedures

The SACM key element is a **CI**. A **CI** is an IT asset or a combination of IT assets that may depend and have relationships with other IT processes.

SACM – Source Code Management inputs element are included in the **Release Package** defined in Release and Deployment Management procedure.

Main input elements are:

- Source Code and Configuration files;
- Archiva configurations;
- Libraries binaries and COTS;
- Patching scripts.

The complete text of the SACM could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.

Annex E - Appendix E

Event & Incident Management

This procedure is applicable to operational activities related to all business services, and information and communication technology assets described within the scope of the EMSA ICT Landscape document.

The end goal of the EIM procedure is to distinguish an event from an incident and:

- In case of an event, to monitor, handle and respond to requests from the users and/or notifications from automated mechanisms, making sure Configuration Items and services are constantly monitored
- In case of an incident, to restore the normal service operation of EMSA systems as quickly as possible, minimizing the adverse impact on business operations, and ensuring the required levels of service quality and availability (Key Performance Indicators – KPI).

Whenever an incident is noted and recorded, the authorised help-desk officer (e.g. a Maritime Support Services operator (MSSO) at EMSA) will carry out an initial diagnosis to identify the affected system(s), the incident model and if there is any available instruction (recovery procedure) on how to solve it. During this stage and if so required the help-desk officer or an authorised project officer or the affected application may contact the 24H contractor (telephone and email) of the affected application for assisting with the diagnosis & resolution process (in line with the Service level agreement drawn between EMSA and the 24H contractor).

The parties involved in the resolution of an incident must ensure that incidents are dealt within true business priority order, meaning if more incidents are on-going at the same time, with the same level of prioritization, the order of applying the corrective actions should consider the impact of each incident on the overall operational response of the Agency.

The complete text of the EIM could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.

Annex E - Appendix F Problem Management

The primary goal of the Problem Management (PM) procedure is to identify the root cause of a set of incidents based on information recorded by the incident management process, and to provide solutions to fix the problem(s).

The procedure will be triggered by the identification of a behavioural pattern of severe and/or repeated incidents. This action can be taken by MSS (Maritime Support Services), the Maritime Application Team (MAT) or ICT Infrastructure team.

Problems in the Maritime Applications and ICT infrastructure can be discovered as a consequence of a survey that leads to connect several incidents to a common root cause or to an initial diagnosis.

When for instance one or more incidents, which impact a specific software module or Configuration Item (CI), are not solved by implementing a suitable incident model available to the Maritime Support Services (MSS), the frequency of the occurrences shows that there is an underlying cause that might be unknown.

If the same incident occurs more than 3 times (in particular within a short period), the MSS shall trigger the PM procedure. For instance the outcome of the quarterly EIM (Event and Incident Management) report can be used to identify the most critical/repetitive incidents as trigger for the PM.

The Problem Ticket analysis is then carried out by the MAT and/or ICT Operational Support. The EMSA responsible officer (MAT and/or ICT) will conduct an investigation in order to identify the root cause of the problem and to propose an interim (workaround) or a permanent solution. Support from software developers might be requested at this stage.

The complete text of the PM could be provided to the successful tenderer on request via e-mail following the kick-off meeting of the contract.