

Working procedures and service requirements

IMPORTANT NOTE

Please note that this document is a template. In all specific contracts under the Framework Contract, working procedures and service requirements goes always along the lines described in this template but the final document will be modified and tuned according to each Specific Contract requirements, when needed.

Table of Contents

1. Introduction.....	4
2. Maintenance.....	4
3. Service Desk	6
4. Processes and Management Plans	6
4.1. CHANGE MANAGEMENT	6
4.1.1. Request for Change	6
4.1.2. Review Change Request.....	6
4.1.3. Evaluate Proposed Change	6
4.1.4. Approval of Change.....	7
4.1.5. Management	7
4.2. RELEASE MANAGEMENT & PLANNING	7
4.2.1. Introduction.....	7
4.2.1.1. Scope	7
4.2.1.2. Goal of Release Management	7
4.2.1.3. Release Management Benefits	8
4.2.2. Release Policy.....	8
4.2.2.1. Release Levels	8
4.2.2.2. Release Lifecycle	8
4.2.2.3. Release Version Numbering	9
4.2.2.4. Release Build Environment & Management	9
4.2.2.5. Required Documentation and deliverables	10
4.3. Incident Management.....	11
4.4. Problem Management.....	11
4.5. Service Level Management.....	12
4.5.1. SLM Preparation	12
4.5.2. SLM Execution & Monitoring.....	12
4.5.3. Service Level Reporting	13
4.5.4. SLM Control (Improvement)	13
5. Deliverables	14
5.1. Corrective and Preventive Maintenance	14

1. Introduction

This document defines the services, processes, management plans and deliverables for the Maintenance phase.

2. Maintenance

After transferring the system to EMSA:

- (1) The Contractor keeps the sole responsibility of diagnosing and determining the origin of failures affecting all or part of the System or Products as mentioned in the Tender Specification.
- (2) The Contractor provides telephone and e-mail support to EMSA during normal working hours to advise EMSA on the use of the Software and to receive any other communication related to the system like incidents and/or services requests, and

For the purpose of this project:

- Corrective maintenance is the reactive modification of a software product performed after delivery to correct discovered problems.
- Preventive maintenance is the modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults. The system has to be updated to the most recent versions of the underlying SW implemented.

The maintenance shall be provided during normal working hours (08.00 to 17.00 Lisbon time) on normal working days (Monday to Friday).

Maintenance reaction time is defined in the technical specification of the FWC.

The following definitions are to be taken into consideration:

- Time to acknowledge – the time the contractor is informed of the problem until the contractor provides an initial investigation and analysis of the problem;
- Time to solve – the time the contractor is informed of the problem until the moment the problem is solved and the service is available again to the end user.
- Type of faults:
 - Urgent – Business stopped: All services are unavailable;
 - Critical - Critical Business Impact: Essential service is not available;
 - Standard – Business Impact: A non-essential function or service is not available.

Maintenance is deemed to comprise of all operations necessary to maintain the system in perfect working order, or to restore a defective system or one of its components to perfect working order, inclusive of the costs of travelling and labour.

Maintenance shall involve the following tasks:

- (1) EMSA preparing and sending to the Contractor all documents and additional information at its disposal which the Contractor might reasonably request in order to reproduce, detect and correct errors;
- (2) EMSA performing onsite testing and accepting, when it is reasonable to do so, of new versions or new releases of software, as proposed by the Contractor,
- (3) The contractor developing and installing any preventive corrections as long as it is agreed by EMSA those corrections are necessary,

- (4) The Contractor diagnosing errors or flaws found by the Contractor, EMSA or any other user in the system and making any necessary corrections,
- (5) The Contractor providing EMSA with successive Software releases, service packs and emergency fixes and the relevant documentation; installing New Software version on the existing environments; where necessary, adapting and configuring the environments that were used by the previous version of the Software,
- (6) The Contractor providing telephone and e-mail support to EMSA during normal working hours to advise EMSA on the use of the Software and to receive any other communication related to the system like incidents and/or services requests.

The Contractor should undertake to provide EMSA, upon request, with any remote maintenance service, which the Contractor operates or intends to set up. All terminal connection, usage and communication charges shall be borne by the Contractor.

The Contractor has sole responsibility of diagnosing and determining the origin of failures affecting all or part of the System or Products as mentioned in the Tender Specification.

3. Service Desk

With the purpose of supporting the day to day use of the system the contractor is requested to provide a Technical Help Desk to serve as a Single Point of Contact to EMSA on technical issues. Technical Help Desk shall be available during the maintenance periods.

4. Processes and Management Plans

In order to support the services for corrective and preventive maintenance described above, at least the following Processes and Management plans must be in place:

- Change Management as described in chapter 4.1
- Release Management & Planning as described in chapter 4.2
- Incident Management as described in chapter 4.3
- Problem Management as described in chapter 4.4
- Service Level Management as described in chapter 4.5

4.1. CHANGE MANAGEMENT

Change Management shall be applied to any change in the system. Such changes can be triggered by EMSA, or by other managerial activities including quality improvement activities, risk management activities, etc. Changes requests may be submitted to increase or decrease the functionality of the solution. Technical changes may be required to overcome challenges in design.

All the changes that have impact to the scope, schedule and/or budget of the project are approved by EMSA according to the change management process. No changes will be initiated until properly documented and approved by EMSA.

4.1.1. Request for Change

Each request for a change in the system is described in a Change Request Form. The form describes the objective and the requirements of the change, identify its impacts on the existing system, and gives orientations on its implementation. Each change request form must be approved by EMSA and the contractor.

4.1.2. Review Change Request

Every proposed change is to be formally documented and submitted for approval using a Change Request Form. The Contractor will review the Change, 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.

4.1.3. Evaluate Proposed Change

The contractor shall evaluate the Change Request to determine the potential impact, risk and complexity of the request.

Simple changes that do not affect the contractual obligations can be prioritised, assigned, and activated on the authority of the Contractor.

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.

4.1.4. Approval of Change

The Contractor will review and present recommendations for dealing with the Change Request to EMSA that is always responsible for making the decision of implementing the recommended changes.

4.1.5. Management

Once the necessary approval has been obtained, the Project Plan and contracts, if needed, are revised as required, and the change action is initiated.

4.2. RELEASE MANAGEMENT & PLANNING

This chapter presents the Release Management as well as the rules, procedures and tools to be used and followed while executing release management related activities.

EMSA Release Management is a mandatory and documented internal procedure. Bidders shall comply with this procedure. A high-level view of this procedure is presented in the following chapters.

4.2.1. Introduction

4.2.1.1. Scope

Release Management is the planning, designing, building, configuring, documenting and testing of hardware and software to create a set of components for a production environment. Release Management should be used for minor, major and emergency hardware and software changes and the bundling of multiple changes into a manageable release.

4.2.1.2. Goal of Release Management

The goal of Release Management is the protection of the production environment and its services through the user of formal procedures and checks working closely with the Change Management and Configuration Management processes.

The main objectives are:

- Plan releases in line with requirements resulting from approved changes.
- Build effective release packages for the deployment of one or more changes into production.
- Test release mechanisms to ensure minimum disruption to the production environment.
- Review preparation for the release to ensure maximum successful deployments.
- Deploy the release in line with structured implementation guidelines.

Because other projects/services may be involved in a new Release, or a heterogeneous or distributed environment could be affected, Release Management must ensure that all technical and non-technical aspects of a Release are covered.

EMSA JIRA/Confluence is the mandatory tool for project issue tracking and requirements tracking, Release Management is heavily dependent on Change Management and Configuration Management. Both processes concur to accurately assess the impact of changes being made and to keep the DSL and CMDB accurate and up to date. Without these processes, Releases will not be controllable.

4.2.1.3. Release Management Benefits

Several benefits can be found when using a well-defined and clear Release Management methodology:

- The possibility to plan resource requirements in advance;
- A structured approach leads to an efficient and effective process;
- Changes are bundled together in a release, minimizing the impact on the user;
- Helps to verify correct usability and functionality before releasing it to the users;
- Provides version control and central storage, which ensures correct version use;
- Provides a set of documentation and benchmarks for each release in production;
- Provides clear communication of release status and release risks.

4.2.2. Release Policy

The release policy covers the following aspects of the release management process: release version numbering, release intervals, the use of emergency releases, the release-unit level for each application, and the deliverables for each type of release.

The policy presents the framework based on which major and / or service pack releases are composed and planned. All releases will be coordinated by project management.

This policy applies to all the software being developed in the context of EMSA's projects.

4.2.2.1. Release Levels

The Project Team shall follow a versioning schema, classified into Major, Service Pack and Emergency software fixes with the following characteristics:

- **Major software release upgrades**, normally containing large areas of new functionality, some of which may make intervening fixes to Problems redundant. A major upgrade or release usually supersedes all preceding Service Packs, Patch Releases or Hot Fixes.
- **Service Pack release upgrades**, normally containing small 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 corrections to a small number of known Problems.

Patches 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 planning. The frequency can be revised at the end of each defined period for the following defined period. Every change that does not fall into the frequency defined/planned should be treated as an emergency software fix and be handled accordingly.

4.2.2.2. Release Lifecycle

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, 4 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.
- 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 4 weeks between deployments¹ 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.

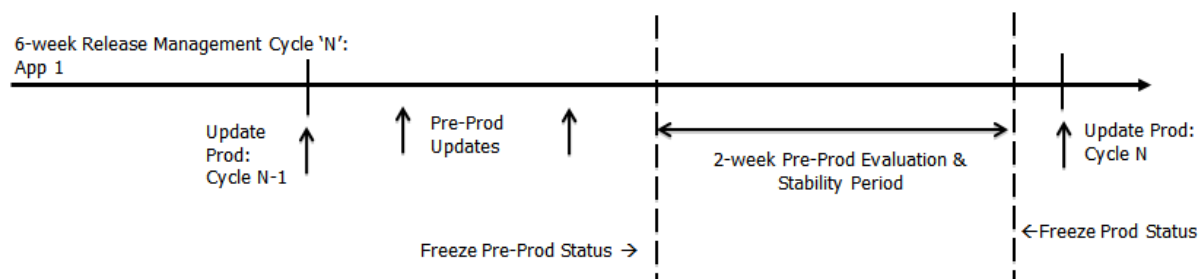


Figure 1 - 6 Weeks Lifecycle

4.2.2.3. Release Version Numbering

According to the release identification mentioned in the previous section, 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.

4.2.2.4. Release Build Environment & Management

<Application/System> building blocks and all their components shall be built at EMSA using EMSA building environment before entering any EMSA environment.

Tools used in the build environment:

¹ Since EMSA closes every year for one and a half weeks during the Christmas period, the preceding days of this period shall be avoided, because the technical support needed for the solving of an eventual problem might not be available. Also the summer vacation period should be taken into consideration when planning for new releases.

Name	Version
Red Hat Enterprise Linux Server	7.4 or higher
Apache Maven	3.3.9
Apache Ant version	1.9.7
Nexus	3.23.0
Jenkins	2.222.3
Sonarqube	7.9.3
Puppet	V2019.11
GitLab Repository	--

Please note that upgrades and patches to these platforms are done with regularity

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 these tools,
- *<Application/System>* (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 Version Control System,
 - All Maven external repositories are configured in the Artifact Management system,
 - All needed components and libraries are downloaded from the external repositories, stored in and served by EMSA Archive,
 - Build procedures are configured in Jenkins,
 - Jenkins build procedure is able to successfully build *<Application/System>* and all related tools/components.
- The release building process is documented and is a part of the release documentation.

4.2.2.5. Required Documentation and deliverables

The following documentation shall be delivered with the new release package:

- Deployment plan (mandatory)
 - Action plan
 - Work breakdown
 - Contingency plans.
- Roll-back plan (mandatory)
 - Identify the “points of no return”
 - Detailed steps required to roll-back to the previous Release
- Configuration scripts (if any)
- For a **full delivery** (typically in the end of the development phase), Contractor should deliver a package containing:
 - source code in a GIT repository
 - binaries for all the COTS and libraries used in the system
 - update/configure archive with the relevant repositories and if needed, manually upload libraries and COTS

- scripts/artefacts/instructions on how to build it
- software configuration item list document, specifying:
 - software configuration items version being delivered
 - COTS and libraries versions
 - release note describing changes
 - reference to the tests made (successful ones of course) and corresponding SPRs (software problem report) solved with the included fixes or new implementations
- installation manual/instructions
- For every updated **delivery** (patch/new release), Contractor should deliver a package containing:
 - source code in a versioning system (currently EMSA uses GIT)
 - binaries for the changed or new COTS and libraries
 - update/configure archive with the relevant repositories and if needed, manually upload libraries and COTS.
 - scripts/artefacts/instructions on how to build it
 - software configuration item list document, specifying:
 - software configuration items version being delivered with instructions specifying what are the differences for this delivery:
 - New components;
 - Changed components;
 - Deleted components.
 - COTS and libraries versions
 - release note describing changes
 - reference to the tests made (successful ones of course) and corresponding SPRs (software problem report) solved with the included fixes or new implementations
 - patch script to update the source code from the current version to the next one
 - installation manual/instructions.

Bidders shall comply with this procedure. Any deviation or change to the above policy or procedure shall be described in the bids.

4.3. Incident Management

Incident Management should include the resolution of incidents and the handling of service requests (e.g. requests for information/support, requests for sending specific notifications to end users) issued to staff offering technical help desk services. The key objective is to guarantee that incidents and requests are handled accurately, completely, and in a timely manner ensuring therefore adherence to the agreed service levels.

Approach

The bidder shall provide a clear approach of the incident management process that includes as a minimum the following activities:

- Incident detection and recording,
- Classification and initial support,
- Investigation and diagnosis,
- Resolution and recovery,
- Incident closure,
- Incident ownership, monitoring, tracking and communication.

4.4. Problem Management

Problem management should include the resolution of problems in response to one or more reported incidents with unknown cause. An appropriate solution needs to be provided for that allows for rapidly and effectively detection and resolving the underlying causes of incidents.

Approach

The bidder shall provide a clear approach to the problem management process that includes as a minimum the following activities:

- Problem Analysis, Categorisation, and Prioritisation,
- Problem Investigation and Diagnosis,
- Provision of the Solution.

A Categorisation and Prioritisation and indication of the Service Level Agreement (at least time to respond and time to correct) shall be defined proposed for each level, based on system specific requirements.

4.5. Service Level Management

Service Level Management (SLM) should be foreseen in order to define and manage the levels of offered services..

The SLM should deliver a cost-effective service that is of quality, meeting or exceeding EMSA's expectations. To this effect, a number of activities are to be performed focusing on:

- Monitoring and analysing the service delivery procedures, performance and quality targets, in terms of compliance to the SLA, effectively and timely reacting to incidents of non-compliance,
- Producing service level reports on regular or on-demand basis (in special cases), which provide a detailed and representative view of the service quality and performance,
- Continuously assessing the procedures and service levels against the nature of events occurred, in order to identify potential problems, deficiencies, trends, etc,
- Suggest and implement improvements.

Service Level Agreements for maintenance shall be defined based on systems specific requirements. An approach on the preparation, and execution & monitoring, and a control (improvement) task related to SLM should include the following details:

4.5.1. SLM Preparation

- The definition of services to be provided to EMSA,
- The definition of service level requirements, criteria, and metrics (Contract, SLA),
- The definition of the on-going activities that support the appropriate service delivery for at least monitoring of metrics, evaluation of service,
- The definition of roles, responsibilities and activities for the services to be offered,
- The identification of potential risks and problems that may be encountered, and the means to handle them.

4.5.2. SLM Execution & Monitoring

- Monitoring and analysing the service delivery procedures, performance and quality targets, in terms of compliance to the SLA, effectively and timely reacting to incidents of non-compliance,
- Producing service level reports on a periodic and on-demand basis, which provide a detailed and representative view of the service quality and performance,
- Continuously assessing the procedures and service levels against the nature of events occurred, in order to identify potential problems, deficiencies, trends, etc.

Mechanisms that will be used for the Service level monitoring and the verification of the compliance to the SLA should be described in detail.

4.5.3. Service Level Reporting

The deliverables of the Service Level reporting should be consistently used for the planning and the implementation of service improvements, for the estimation of the impact of introduction of changes and generally for all the activities related to the management and control of the service.

Formal reporting shall be performed including at least:

- Activities performed during the reporting period,
- Analytical presentation of the service evaluation metrics, along with the corresponding analysis results,
- Comparison between the achieved and the agreed service levels, presented in a way that is both detailed and representative of the service performance and quality,
- Statistics on the number of requests made, the number of open and closed calls, the needed average time taken to initiate an action, and the average time taken to resolve incidents,
- The key issues identified through the service delivery during the reference period. With the term “key issues” we refer among others to:
 - Encountered problems related to either the performance of the Contractor, or the performance of other parties affect the service delivery,
 - Identified trends and changes on the service environment that could be introduced by various factors,
 - Points of improvement and directions for future evolution of the service,
 - Handling of special cases (for example an unusual fluctuation on the experienced workload).

A summary of the planned actions to be performed by the service personnel during the next reporting period.

Suggestions proposed concerning the resolution of identified problems and/or the improvement of the service in terms of quality and/or cost-effectiveness.

Planning for the next reference period.

4.5.4. SLM Control (Improvement)

The Contractor should foresee a process that manages its quality control of the service, and specifically to the extent in which the Contractor has succeeded in designing services and achieving targets according to EMSA's requirements. This process does not refer to the achievement of the service levels, which is part of the SLM monitoring processes described in the previous sections. The SLM control process is to record the effectiveness, performance, and quality of the procedures and mechanisms applied, to identify the cause(s) of poor performance and propose appropriate changes and improvements. The results of the control process are to be used as an input for performing necessary changes to the Project Management and Quality Control Manual under the agreed review procedures.

5. Deliverables

5.1. Corrective and Preventive Maintenance

The deliverables that are to be produced in the context of Maintenance activities are:

- Monthly (or any period upon agreement) Maintenance Reports and Statistics on maintenance activities described in the context of Service Level Management.
- Change Management Documents for each change submitted to the Change Management Process. It must include at least, Change Request Form, Evaluation of the Change, Planning and Acceptance
- Updated versions of the system deliverables (design documentation, test documentation, user documentation, system documentation, software releases and release notes) for each change implemented and submitted to the Release Management Process.

Reports shall be delivered to EMSA as specified in the technical specification of the FWC.

The meetings in which these reports and statistics are discussed are specified in the technical specification of the FWC.

European Maritime Safety Agency

Praça Europa 4
1249-206 Lisbon, Portugal
Tel +351 21 1209 200
Fax +351 21 1209 210
emsa.europa.eu

