

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, test<sup>1</sup>, 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

---

<sup>1</sup> 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.

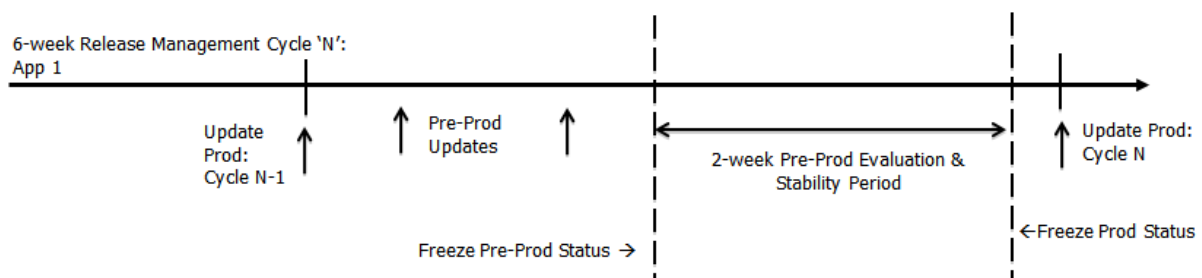
When planning new releases, the following guidelines should be taken into consideration:

- 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.