

# **EMSA Guidelines for Requirement Definition**

**November 2020**

**Document Approval**

	Name	Date
Prepared by:	Mario Oliveira	12/11/2020
Quality control by:	ICT SG	23/11/2020
Approved by ICTSG:	Peter Kirov (Chairperson)	23/11/2020
Owner	Peter Kirov	23/11/2020

**Change Control History**

Version	Date	Description	For Info / Approval
1.0	August 2016	First version presented to HoU C.4	Info
1.1	September 2016	First version presented to HoD C (CC to HoUs C.1, C.2 and C.3)	Info
1.2	October 2016	Second version containing inputs/comments from C.1, C.2, C.3 and C.4	Info
1.3	November 2016	Third version containing suggestions from the open session conducted with C.1, C.2, C.3 and C.4	Approval
2.0	May 2018	New version containing feedback received and suggestions provided during the open session conducted with C.1, C.2, C.3 and C.4	Approval
3.0	November 2020	New version adjusted to the new EMSA JIRA/Confluence tool and containing feedback received and suggestions provided by CAB, TPMs and BPMs.	ICT SG Approval

**Document information**

Filename	EMSA Guidelines for Requirement Management_v3.0.
Location	EMSA - Lisbon
Number of pages	20

---

# Contents

---

Contents ..... 3

I. Introduction and Objective ..... 4

II. Common Terminology and Definitions ..... 5

III. General considerations to take into account when defining requirements ..... 7

IV. Conclusions..... 10

    Annex 1 – Techniques for the Requirement Elicitation ..... 11

    Annex 2 – Requirement drafting Example..... 12

    Annex 3 – Requirements definition as implemented in JIRA/Confluence ..... 13

    Annex 4 – Use Case ..... 18

---

## I. Introduction and Objective

---

Requirement Definition (RD) is increasingly recognised as a critically important activity in any systems engineering process. Effective RD plays a key role in determining the success or failure of projects, and in determining the quality of systems that are delivered.

RD is a multi-disciplinary, human-centred process, concerned with identifying stakeholders and interpreting and understanding their needs. It is often regarded as a front-end activity in the systems development process. However it is usually also the case that requirements change during development and evolve after a system has been in operation for some time. Therefore, RD plays an important role in the change management process as well.

Nevertheless, the bulk of the effort of RD does occur early in the lifetime of a project, motivated by the evidence that requirement errors, such as misunderstood, omitted or poorly-defined requirements, are more expensive to fix later in project lifecycles. The assessment of a project's feasibility and associated risk needs to be undertaken, and RD plays a crucial role in making such an assessment.

Therefore it has been identified as essential to establish best practices and templates for the necessary RD activities.

The objective of this document is to provide guidelines for reference when defining requirements for EMSA maritime applications and corporate ICT systems, which are subject to software development and/or configuration control. Therefore it provides the necessary rules to define, document and manage the requirements.

These guidelines are designed as a reference and do not, pretend to be exhaustive. Moreover the present document may need to be adapted to particular situations according to the experience and competence of the staff involved<sup>1</sup>. This document does take into account existing EMSA procedures such as the ones approved by the ICT SG (e.g. Change Evaluation Management, Service Level Management, Service Validation Verification and Testing).

These guidelines should be used by all Business Project Managers (BPM) and Technical Project Managers (TPM) in order to achieve a consistent quality method.

It is important to take into account common terminologies and definitions for the Requirement Definition and these are listed in the following sections.

---

<sup>1</sup> Exceptionally, requirements concerning simple ICT projects that do not impact any other application/service can be registered and managed differently if previously approved by the HoU.

---

## II. Common Terminology and Definitions

---

The following are a set of common definitions used throughout these Guidelines.

**Actor:** A person, organisation or system that interacts with the system being analysed.

**Attribute:** Attached data inherent to individual requirement that must be collected and updated in order to document extra information that allow understanding and managing requirements (e.g. priority, status, acceptance criteria, etc).

**Confluence:** web-based corporate wiki that integrates seamlessly with Jira to provide context to Jira issue

**Functional Requirement:** Condition or capability that explains what the system must do describing a specific behaviour or function of the system.

**High Level Requirement:** Business requirements defined by service managers before splitting them in atomic requirements which will be then used in one or several Request For Changes (RFC).

**Issue:** record created and managed in JIRA.

**Issue Status:** current status of an issue in JIRA based on the workflow transitions.

**JIRA:** family of web-based products designed to support teams of all types to manage work. It includes modules for software development and helpdesk/customer relationship management.

**Non-Functional Requirement:** Condition or capability that describes a quality goal/constraint of the system that is going to be developed. Non-Functional requirements could be related to one or more functional requirements.

**Requirement:** Condition or capability that must be met or possessed by a system to satisfy a contract, standard, specification, or other formally imposed documents.

**Requirement Analysis:** Iterative sub-process which takes place after the elicitation of the requirements, involving the various stakeholders to break down functional and non-functional requirements to a basic understandable view to provide a clear system development process framework.

**Requirement Definition:** Discipline which includes the requirement elicitation plus the action to write and then how to format and to store the requirement.

**Requirement Elicitation:** Practice of collecting the requirements of a service and/or system from users and other stakeholders, which includes interviews, questionnaires, user observation, workshops, brainstorming, use cases, role playing, prototyping, etc.

**Requirement Engineering:** Process of collecting, analysing, refining, and prioritizing system and stakeholders' needs, agreeing on requirements and then controlling change and communicating to relevant stakeholders.

**Requirement Management:** Technique used to ensure that both traceability and change management are properly applied.

**Requirement Modelling:** Technique that uses a combination of text and diagrammatic forms to depict requirements in a way relatively easy to understand.

**Requirement Specification:** Also known as Requirement Description is a detailed statement of a user need and/or of the properties that a system is required to achieve normally produced in response to a needs, and then used as the basis for system design.

**Requirement Validation:** Iterative sub-process which takes place throughout the lifecycle of the project aiming to ensure that each requirement specification and its attributes are complete, consistent, modifiable and traceable.

**Service:** A means of delivering value to users to satisfy a need or to fulfil a demand.

**Stakeholder:** Individual or organisation interested in the success of a service or system.

**System:** An integrated composite that consists of one or more of the processes, hardware, software, facilities and people, that provides a capability to satisfy a stated need or objective.

**System Requirements:** All of the requirements at the system level that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements, and are expressed in an appropriate combination of textual statements, views, functional and non-functional requirements.

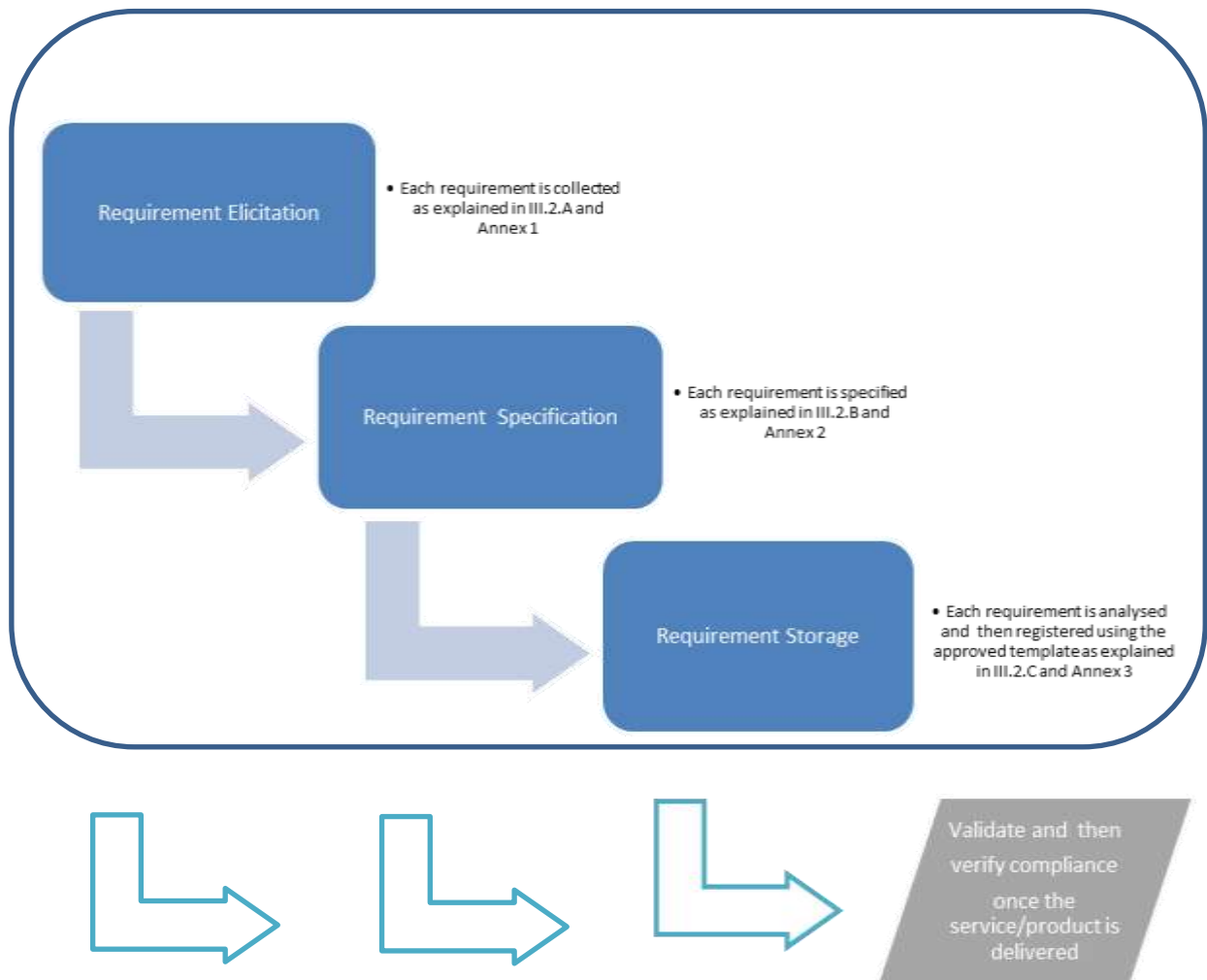
**Use Case:** Methodology used in service/system analysis to identify, clarify, and organize service/system requirements, typically to define the interactions between an actor and a system, to achieve a certain goal.

**Use Case Diagram:** Technique used to represent a system scope showing some of the functions of the system (functional requirements) and external entities that interacts with them.

**User:** Entity that has the right to use a system or service to obtain benefits from it.

### III. General considerations to take into account when defining requirements

#### Overview of the process and sub-processes



### III.1 Applicability

These guidelines are applicable to all BPMs and TPMs involved in the development and management of system/service requirements. The rules and templates presented should be applied for the specification of new requirements and also for the review of existing ones.

### III.2 Rules for defining requirements

#### **A. Requirement Elicitation** (how to collect the requirements from users and other stakeholders)

First of all is essential to identify who are the stakeholders related to the project and to maintain a generic list of stakeholders during the all cycle.

It is also important to inform other BPMs of envisaged developments in order to identify synergies between the needs of the different user communities and agreeing on the best technical solutions and implementation priorities according to available budgetary and staff resources. The Integrated Maritime Services the Design Coordination Group (DCG IMS) was established<sup>2</sup> to coordinate the design and the implementation of new or improved integrated maritime services.

There are several techniques for collecting requirements and ensuring their completeness. These are described in Annex 1.

Business requirements collected from end users are often high level requirements that will result after analysis into one or several atomic requirements during the specification phase. With this approach, user (business) requirements can be kept as specific as possible, but still refer to use cases.

This step of Requirement Elicitation should not be considered as having a dedicated time as it will depend on the type of project. For the existing applications/services it is conducted continuously and each set of requirements, elicited during a certain period, should be then collected to form a specific Request For Change (RFC). All requirements shall be registered and continuously managed in JIRA/Confluence using the electronic templates and workflow specifically created for this activity. The same is happening with the management of the RFC, as detailed in the Change & Evaluation Management (CEM) procedures.

#### **B. Requirement Specification** (how to draft requirements)

The draft of the requirements is vital for the all process. It is essential to know that requirements must be written in a language that is understandable by those that must read them. The main objective is to ease the reading and comprehension of the requirements.

The most common method to write requirements is to use natural language. However in some situations/scenarios it could be advantageous to apply Use Cases descriptions and diagrams to define requirements too.

---

<sup>2</sup> Decision No. 2020/007 of the Executive Director relating to the establishment of the Design Coordination Group for Integrated Maritime Services



Use Case descriptions and diagrams should be used<sup>3</sup>, when practical, as a technique to represent a system or a service showing its functions and the external entities that interacts with them. EMSA Contractors should be also advised about the use of Use Cases descriptions and Use Case diagrams to translate high-level business requirements.

The way how to describe a Use Case is shown in Annex 4.

During the specification or description of requirements it should be taken into consideration that they must be:

- **Atomic:** it specifies a single need or characteristic, and is focused on transmitting a single aspect, without using conjunctions such as “and” / “or”<sup>4</sup>;
- **Clear:** it is easy to read and understand, using short, simple, affirmative and concise sentences;
- **Non-ambiguous:** it has a single and objective interpretation;
- **Design independent:** it defines service and/or system characteristics that will be observed externally (“what is wanted”) and not the internal details about how it will be implemented that could inadvertently force a technical design;
- **Feasible:** it is technically assessed to make sure that is executable with the technology, time, and budget available;
- **Traceable:** the requirement must be identifiable and traceable to the design elements that implemented it, if applicable, and to the test cases that validate fulfilment thereof.

Examples of requirements following these principles can be found in Annex 2.

After applying these rules the full set of requirements has to be revised first with end-users, if applicable, and then with domain experts, in order to ensure that it is complete, consistent, modifiable and prioritised. This action called “peer review” should be performed by the business and technical leads of the service which will be most impacted by the RFC in question and take place just after the first draft of the RFC. By doing this activity the reviewers should also confirm that the set of requirements are complete, consistent, modifiable and prioritised.

The TPM shall, with the support from the BPM, confirm that these rules were followed before classifying the requirement(s) as ready to be submitted (i.e. “final”).

It should be noted that this step of Requirement Specification needs to be performed in a contained period of time in order to have the RFC ready to be reviewed in time for Change Advisory Board (CAB) assessment and HoU/HoD approval<sup>5</sup>.

---

<sup>3</sup> Training on how to draft Requirements and apply Use Case descriptions and diagrams was already done for some staff members. This training is part of EMSA Training Catalogue.

<sup>4</sup> It should be taken into consideration that in some exceptions could require the use of “and” / “or”.

<sup>5</sup> As a generic role the Requirement Specification should not take more than one month.

### **C. Requirement Storage** (how to format and store requirements)

Each requirement must be properly analysed, validated, structured and stored in JIRA/Confluence, in the relevant project space using the electronic template created in JIRA/Confluence and shown in Annex 3<sup>6</sup>.

Once stored in JIRA/Confluence the rest of the lifecycle of the requirement(s) must be handled by the owner using the workflow implemented in the tool (i.e. Draft, On Hold, To be Reviewed, Cancelled, Final, In Progress, Delivered, Validated, In Production, Retired) as explained in Annex 3.

To be noted that there is an iterative sub-process, named “Requirement Validation” which takes place throughout the lifecycle of the project, which aims to ensure that the requirement specification is complete, consistent, modifiable and traceable. As shown in the process overview diagram the compliance of the service/product delivered should be also checked against the requirement defined.

It should also be noted that the CAB for assessing the RFC and HoU/HoD for approving them need an overview of the RFC in a single webpage and/or document<sup>7</sup>.

---

## **IV. Conclusions**

---

This document presents above (in section III.2) the guidelines for defining requirements, following three main aspects:

### **Requirement Elicitation**

### **Requirement Specification**

### **Requirement Storage**

These Guidelines are aimed for all EMSA staff dealing with ICT related projects.

The document will be updated as needed, at least once a year, although templates could be updated more regularly if required without the need for formal circulation. Therefore, all staff involved on the management of requirements are advised to refer and use these templates on a regular basis, which are electronically implemented in JIRA/Confluence.

---

<sup>6</sup> In order to be able to present the requirements in the right logical order, BPMs should include an external reference ID as described in Annex 3.

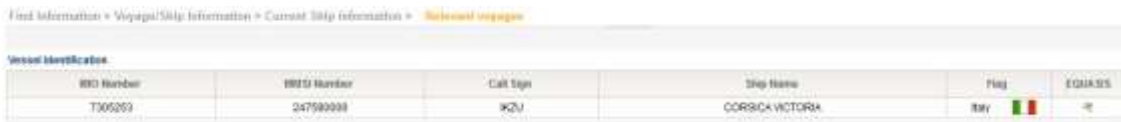
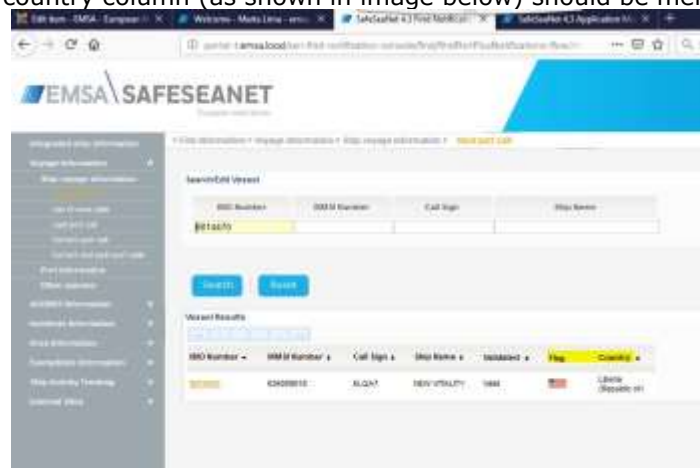
<sup>7</sup> View automatically generated in Confluence as soon as the RFC is status is set to ON HOLD.

## Annex 1 – Techniques for the Requirement Elicitation

The following techniques can be used in Requirement Elicitation (list not exhaustive):

- Interview
- Questionnaire
- Brainstorming
- Change of perspective
- Perspective-based reading
- Reuse of requirements
- Field observation
- Apprenticeship
- Workshops
- Prototypes
- Recording
- Use case modelling

## Annex 2 – Requirement drafting Example

Title : Merge of Flag and Country columns on web interface				
Issue JIRA Number: EIS-110		Requirement Order ID in RFC : n.a.	External Reference: n.a.	PRIORITY: Must have
SERVICE: SafeSeaNet			Type: FUN - Functional	
SUB-SERVICE: n.a.				
APPLICATION: EIS			COMPONENTS: CORE, GUI	
<DESCRIPTION>				
<p>On the web interface where there is the Flag and the Country displayed in the results, these should be displayed as a single column named "Flag" as presented in the figure below.</p>				
				
<p>For example in the “Find information” module, in the “Vessel Identification” section, the flag and the country column (as shown in image below) should be merged into one column Flag</p>				
				
STATUS : Logged			PEER REVIEWER	limagma
ACCEPTANCE CRITERIA			<p>Login to the web interface of EIS and in the following sections check that the Flag and Country columns are merged:</p> <ul style="list-style-type: none"><li>- Find information : Search vessel</li><li>- Find information : Vessel identification</li><li>- Find information : Integrated ship information</li><li>- Find information : Latest AIS/MRS for a selected ship</li><li>- Incidents information : Vessel Identification Criteria</li><li>- Find information : Ship Activity Tracking</li><li>- Send Notifications : Exemptions on scheduled services</li><li>- Vessel Management console : Vessel Management</li></ul>	

## Annex 3 – Requirements definition as implemented in JIRA/Confluence

Title			
Issue JIRA Number: <project number>	Requirement Order ID in RFC:	External Reference:	PRIORITY: *
SERVICE: *		Version:	
APPLICATION:		Type: *	
SUB-SERVICE: <sub-service>			
COMPONENTS: <components>			
DESCRIPTION: * <i>(Note: it could be done using Use Cases)</i>			
STATUS: *		PEER REVIEWER:	
ACCEPTANCE CRITERIA: * <i>(not necessary for atomic requirements)</i>			

\* Mandatory field.

NB: for describing associated RFC and Service Requests as well as dependencies (parent and/or child requirements) the already existing functionalities of JIRA/Confluence (i.e. "Links") should be used.

#### Definition of the Requirement Template's fields

Service	Text from list of services below	
JIRA ISSUE NUMBER	Number attributed automatically by JIRA	
Version	Requirement version number	
Requirement Order ID in RFC	X...X_ZZZ	X...X – Code with a reduced number of capital letters identifying the change requested
		ZZZ order number used to sort automatically the requirements when generating RFC pages in confluence for CAB assessment
External Reference (to be used just if required)	X...X_YYY_Z...Z	X...X – Code with a reduced number of capital letters identifying the requesting service (see table below)
		YYY – three capital letters identifying the type of requirement (see explanation below)
		Z...Z – unique number of requirement (3 to 5 numerical characters created by the business team)

REQUIREMENT TYPE (categorisation of each requirement defining its type)	FUN	Functional
	CAP	Capacity (Non-Functional)
	PER	Performance (Non-Functional)
	AVA	Availability (Non-Functional)
	CON	Continuity (BCF) - (Non-Functional)
	SEC	Security (Non-Functional)
	TEC	Technical (Non-Functional)
	MON	Monitoring (Non-Functional)
	DOC	Documentation (Non-Functional)
	REL	Reliability (Non-Functional)
	SCA	Scalability (Non-Functional)
	TES	Tests (Non-Functional)
	INF	Informative
	HLV <sup>8</sup>	High Level Requirement
	OTH	Other
PRIORITY (prioritise the level of importance of each requirement following MOSCOW principle)	Must have (i.e. Cannot deliver without this)	
	Should have (i.e. Important but not vital)	
	Could have (i.e. Desirable but not important)	
	Won't have (i.e. Agreed that it will not be delivered)	
SERVICE (identify the relevant service(s) related to the requirement)	select one of the the service(s) from the table 1 below, which follows the approved EMSA Service Catalogue	
SUB-SERVICE (if applicable, identify the relevant sub-service(s) affected by the requirement)	"text" with the name of the sub-service	
APPLICATION (identify the relevant application related to the requirement)	"text" with the name of the application	
COMPONENTS (if applicable, identify the relevant components affected by the requirement)	"text" with the name of the component(s)	
DESCRIPTION (describe each requirement in a clear way following the rules explained in paragraph III.2.B)	"text" with the description of the requirement	
STATUS	Draft	Requirement under preparation

<sup>8</sup> This type of requirement is not possible to use when the requirement is submitted to the CAB.

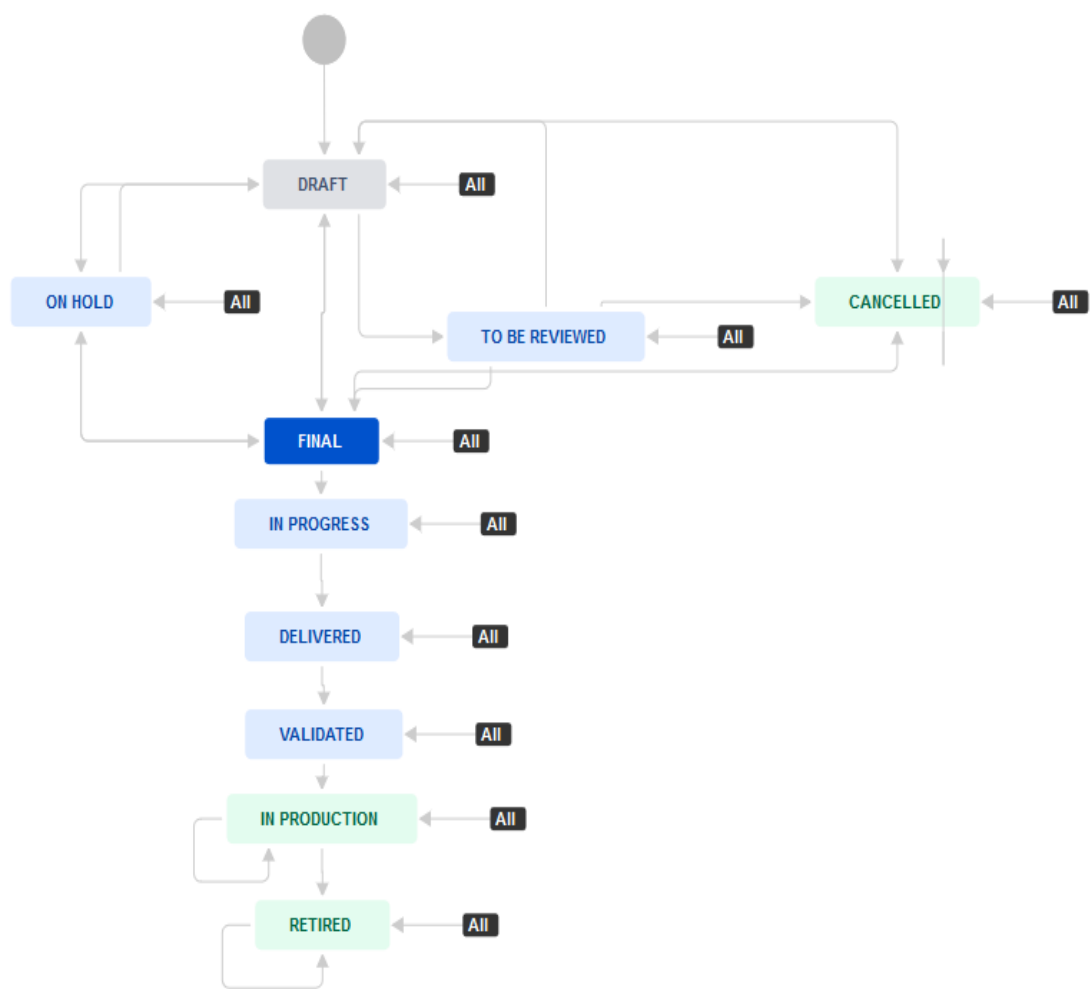
(current status of requirement) – see table 2 with the workflow below	To be reviewed	Requirement being reviewed (e.g. by peer review)
	On Hold	Requirement drafted and submitted to the Change Advisory Board (CAB)
	Cancelled	Requirement never went into operation and is no longer applicable
	Final	Requirement assessed by the Change Advisory Board (CAB)
	In progress	Requirement approved by HoU (or HoD) and in progress for implementation
	Delivered	Requirement implemented and ready to be tested and validated
	Validated	Requirement tested and validated, and ready to go into production
	In Production	Requirement implemented in production
	Retired	Requirement was operational but has been phased out / replaced
PEER-REVIEWER (define one person to execute a peer review and validate the requirement)	Person name – date is automatically set by JIRA	
ACCEPTANCE CRITERIA (Default: Acceptance Criteria must be included unless an exception can justified & described in the text box, for instance for atomic requirements)	"text"	

**Table 1 - List of Services  
(taken from approved EMSA Service Catalogue)**

EMSA IMS-EUNAVFOR  
EMSA IMS-EUNAVFOR-MED  
EMSA -IMS EFCA  
FRONTEX  
EMSA IMS-MAOC (N)  
EMSA IMS-Europol  
SAFEMED IV  
Preparatory measures for future participation of relevant IPA II countries in EMSA  
BCSEA project  
IMS for Member States Authorities  
SEG – SafeSeaNet Ecosystem Graphical User Interface  
IMS Mobile App  
STAR ABM (Automated Behaviour Monitoring)  
LONG TERM STORAGE (LTS, part of HP-IMS)  
STAR TRACKING  
Satellite AIS (SAT-AIS) data services  
STAR STREAMING (Sat- AIS Data distribution)  
Dynamic Search Aggregator (DSA)  
Integrated Reports Distribution (IRD)  
Ship Tracking, Awareness and Reporting data – Real-Time Maritime Picture Service (STAR RTMPS)  
TDMS  
Ship Tracking, Awareness and Reporting data – VDS Correlation Service (STAR VDSCS)  
Business Intelligence – IMS, SSN, and analytics use cases  
Maritime Analytics tool - prototype  
Central Ship Database (CSD)  
Central Geographical Database (CGD)  
Central Organisation Database (COD)  
Central Location Database (CLD)  
Open Geospatial Consortium Central Location Database (OGC-CLD)  
Central Hazmat Database - CHD  
MAR-CIS  
CleanSeaNet (CSN)  
Copernicus Maritime Surveillance (CMS)  
SafeSeaNet European Index Server (SSN EIS)  
STAR STREAMING (T-AIS data)  
SSN SI  
European Union Long-Range Identification and Tracking System Cooperative Data Centre (EU LRIT CDC)  
International Long-Range Identification and Tracking System Data Exchange (LRIT IDE)  
European Marine Casualty Information Platform (EMCIP)  
MED Database – ‘Marine Equipment Directive Database’  
THETIS PSC  
THETIS EU – MARSEC  
THETIS EU – PRF  
THETIS EU – SRR  
THETIS EU - RoPAX  
THETIS EU - Sulphur  
THETIS eCertificates  
THETIS Med  
THETIS MRV  
STCW-IS: Standards of Training, Certification and Watchkeeping Information System  
Maritime Knowledge Centre Services (MaKCs)  
Virtual Reality Environment for Ship Inspections (VRESI)  
Rulecheck  
Maritime Support Services  
RPAS Services  
Identity Management Common Management Console (IDM CMC)  
Corporate Services  
Other



Table 2 - JIRA Requirement issue workflow



## Annex 4 – Use Case

Use Cases are a technique for capturing, modelling and specifying the requirements of a system. A Use Case corresponds to a set of behaviours that the system may perform in interaction with its actors, and which produces an observable result that contribute to its goals. Actors represent the role that human users or other systems have in the interaction.

Use Cases belong to the scenario-based requirement elicitation techniques, as well as the model-based analysis techniques. But the Use Cases also supports narrative-based requirement gathering, incremental requirement acquisition, system documentation, and acceptance testing.

The description of a Use Case can include:

- Name (recommended to be a verb and a direct object)
- Objective
- Actors
- Event/Trigger (what triggers the execution of the use case)
- Precondition (state in which the service/system must be in order that the use case can be executed)
- Normal flow (set of steps of the normal scenario describing the flow of execution in the most usual case)
- Alternative flow (set of steps of the normal scenario describing the flow when something happens disrupting the normal flow)
- Post-condition (final state of the system after the use case is executed)

A Use Case is used to describe a scenario of interactions between two or more actors, through a sequence of numbered steps and the following best practices should be applied for the description:

- Use Active Voice (an actor or the system must always be the subject of the sentences)
- Write simple and complete sentences (each step must have a subject (actor or system), a verb (function) and a predicate (at least a direct object))
- Keep the correct level of detail (only to be included if the user sees this)
- Be consistent in names of actors and objects (using the same names in all scenarios)
- Be clear in the identification of the end of the use case
- Do not add description of internal logical rules to the use case (the use case focus is only the interaction between actor and system)

An example of a Use Case is shown below.

Use Case Req ID	<b>UC-ROL-ADD-ANN</b>	
Use Case Name	<b>Assign Role Announcement</b>	
Purpose	Covers the functionality related to the system's actions when notified of a user's role assignment from the IdM Provisioning Service.	
Primary Actor(s)	External system (OIM/IdM Provisioning Service)	
Precondition(s)	"SafeSeaNet" application must be associated in OIM to the user being provisioned.	
Postcondition(s)	1.The system returns a SOAP response informing about the result of the "Assign Role" process. 2.The system adds the user's role in SSN database.	
Trigger(s)	A "SafeSeaNet" application role has been assigned to user using OIM.	
Use Case Description	<b>Primary Workflow</b>	
Step 1	OIM has effectively sent information on user's role assignment by invoking "assignRole" WSDL operation of IdM Web Service.	
Step 2	The system checks whether the reported role is already assigned to the resolved user.	
Step 3	The system updates resolved user's roles in SSN database by adding the reported role and returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_ADD_ROLE_SUCCESSFUL".	
Alternative Use Case Description	<b>S2A1:</b> No IdM user is found in SSN database for the "userId" specified in the "AssignRoleMessage".	
Step 1	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="USER_DOESNOT_EXIST".	
Alternative Use Case Description	<b>S2A2:</b> Reported role is already assigned to the resolved user.	
Step 1	The system returns a SOAP response with payload an "IDMResponse" with StatusCode="DUPLICATE_ROLE".	
Input(s)	"AssignRoleMessage" WSDL message.	
Output(s)	The reported user's role added in SSN database.	
Timer(s)	-	
Business Process(es) Reference	-	
Associated Use Case(s)	-	
Special Requirements	-	

The use case diagram is an overview of a system or a service showing its functions and the external entities that interacts with the respective functions.

An example of a Use Case Diagram is shown below.

