

New EMCIP

Project Delivery

Appendix I of EMSA/OP/15/2016, “Development of New European Marine Casualty Information Platform (NEW EMCIP)”

v.1.0

Date: 13/06/2016

Table of Contents

1. Project Phases	3
1.1. Initiation.....	3
1.2. Design	3
1.3. Development and Test.....	4
1.4. Deployment	4
1.5. Training	4
1.6. Go-Live.....	5
2. Project Deliverables	5
2.1. Initiation.....	5
2.1.1 Project Management documentation	5
2.1.2 Software Development Plan	5
2.1.3 Project Tools	6
2.2. Design	7
2.2.1 Design Documentation	7
2.3.1.1. Operational and Maintenance Documentation	8
2.3.1.2. System building procedures	8
2.3.1.3. Infrastructure (HW and SW) documentation	8
2.3.2 Functional Prototypes	9
2.3.3 Final Version	11
2.3.4 User Documentation	11
2.3.4.1. User documentation	11
2.3.4.2. Training support material	11
2.3.5 Test Documentation.....	12
2.4. Deployment and Data Migration	12
2.5. Go-Live.....	13
2.6. Training	13
3. Timetable.....	13
4. Acceptance Procedures	14
4.1. Classification of Software Issues	14
4.2. Documentation	15
4.3. Final System	15
4.4. Final System	15
4.5. Quality Gate	16
5. Meetings.....	16
5.1. Project Management Meetings	16

This document details the phases required for the New EMCIP project delivery, from the initiation and design of services until final acceptance of the system, excluding warranty.

This document presents all the specific project delivery phases and it describes what deliverables are requested, when they should be delivered and how they will be accepted by EMSA.

1. Project Phases

Six major phases are foreseen from signature of the contract until the final system is accepted, which are:

- Initiation;
- Design;
- Development and Test;
- Deployment;
- Go-Live;
- Training.

Deliverables are expected for each phase. They are detailed in Chapter 2 – ‘Project Deliverables’ of this document.

If one deliverable is not accepted by EMSA during one phase no acceptance of any deliverable of the following phase can be done.

1.1. Initiation

The objective of this initiation phase is to have a mutual understanding and agreement of methods and means that will be used for the completion of the project.

Immediately after the signature of the contract the contractor should prepare the kick-off meeting to cover at least the following subjects:

- objectives and organization;
- contractor team;
- project tools;
- project plan;
- methodologies and procedures;
- software development plan;
- content and level of detail of the project management documentation.

During this phase, the contractor is asked to work in close contact with EMSA in order to create a common view of the whole project.

1.2. Design

The objective of the design phase is to create a complete set of functional and technical specifications specifying what and how is to be implemented, the methodologies that shall be used to verify and validate the project execution, and setting down the Migration Strategy for migrating data from the existing EMCIP to the New EMCIP.

1.3. Development and Test

The objective of this phase is to develop and test the final version of the system before deployment and import of data.

The application will be developed according to the deliverables of the design phase.

Before delivery the contractor must test the developed application to verify the conformity with expected results and validate that the procedures as stated during the previous phases have been applied. The contractor should respect the "two chamber principle" which means that the team in charge of the tests should be different from the team in charge of the design and development. Tests cannot be executed before prior acceptance of Test documentation by EMSA. Test results should be transmitted to EMSA.

During this phase the contractor is responsible to:

- Prepare system documentation;
- Deliver a final version of the system;
- Prepare User documentation, including training support material;
- Prepare Test documentation;
- Test final version before its delivery and report test results to EMSA;
- Prepare and implement the migration procedures and scripts, according to the defined Migration Strategy;
- Test the migration procedures and scripts.

1.4. Deployment

Deployment starts as soon as all deliverables of the development phase are delivered to EMSA.

The objective of the deployment phase is to configure, migrate the data and make the final version available and fully running on its environments:

- Test;
- Pre-Production/Quality;
- Production.

During this phase EMSA will perform acceptance tests to accept the final version, the system documentation, the user documentation and the training materials.

1.5. Training

The training phase starts after acceptance of final version, user documentation and training materials.

During this phase the contractor should organise and conduct the following training session:

- teach EMSA's operational unit end-users to use the application;
- teach EMSA's IT personnel to manage and administrate the system.

Training will be done on the test and/or quality environment.

In case adjustments or corrections are found necessary during training sessions, the contractor will be asked to update user documentation and training materials accordingly.

1.6. Go-Live

The go-live phase starts after the final version is accepted by EMSA. The objective of this phase is to:

- obtain an optimum configuration of the system and maximal performance in the production environment by fine tuning the complete technical infrastructure;
- perform necessary correction and adjustments of the system while it is used by end users in real situation.

The go-live phase ends at the final acceptance of the system.

2. Project Deliverables

Documentation must be provided in electronic format compatible with MS Office 2010 or equivalent.

2.1. Initiation

2.1.1 Project Management documentation

Project management documentation should reflect the project management methodology proposed by the contractor in its bid. It should include at least the following documents:

- Project charter: details the understanding of the project, the different methodologies to be used and the first project baseline (in line with the first project plan);
- Project plan: must include the following items at least: project charter, project management approach, scope, Work Breakdown Structure (WBS), project team, Gantt chart, deliverables milestones, working locations, meetings planning and reports;
- List of outstanding and closed Action Items;
- Flash report: simple report on the status on the project containing (at least) ongoing tasks, resources usage, progress status, and issues foreseen;
- Agenda of the meetings: the contractor is responsible for providing detailed agenda and additional requests 3 days before the meetings for all relevant meetings held between EMSA and the contractor;
- Minutes of the meetings: the contractor is responsible for providing the minutes of the meetings for all relevant meetings held between EMSA and the contractor. The minutes of the meetings must include at least the topics discussed, decisions taken and action items with indication of the responsible person and deadline of the actions.

2.1.2 Software Development Plan

The Software Development Plan identifies the general methodologies, processes and working practices to be used on the development of the project. It serves as the basic rules and practices guideline for all the remaining technical tasks of the project. The Software Development Plan will at least address the following issues:

- Software development approach: description of the strategy of the software development life cycle

(waterfall, incremental, evolutionary life cycle, etc);

- Software engineering environment: identification of any standards, methods and tools that apply to the project, whether defined by the contractor or required by EMSA. Details regarding the application of each item identified shall be addressed on the subsequent sections when referring to them;
- Software quality assurance process: definition of rules, practices and conventions to be used as to obtain the desired quality of the final product and to evaluate if this plan is being properly implemented. References to standards or one or more contractor's own documents are possible;
- Software configuration management plan: definition of rules, practices, conventions and tools to be used as to maintain the software configuration. References to standards or one or more contractor's own documents are possible;
- Design standards: definition of rules, practices, conventions and tools to be used in definition of the design. References to standards or one or more contractor's own documents are possible;
- Coding standards: definition of rules, practices, conventions and tools to be used in development of the code. References to standards or one or more contractor's own documents are possible;
- Testing standards and practices: identifies the standards, practices, conventions and tools that will be used to test the developed software. References to standards or one or more contractor's own documents are possible.

2.1.3 Project Tools

Unified Modelling Language (UML) should be used for object and system modelling. The UML modelling tool and any additional tools suggested by the contractor in its offer will be used.

EMSA suggests the use of an Open Source UML tool, like "ArgoUML. Another modelling tool can be used if agreed with EMSA and as long as EMSA has access to it without licencing costs. Input files to the UML tool must be provided..

Project progress controlling method will be proposed by the contractor. EMSA suggest EVM (Earned Value Management).

The contractor will provide a project site accessible by EMSA through internet. The project site should contain, at least:

- Last stable version of all document deliverables;
- Last stable version of the prototype or system;
- Current working documentation;
- Current development snapshot.

The documentation provided by the contractor shall be suitable for Microsoft Office 2010 suite (or higher version).

The project site must provide version control of all documentation and source code, making possible to retrieve at any time previous versions and the last stable version of source codes and documentation.

2.2. Design

Design documentation should be prepared in close collaboration with EMSA's personnel.

2.2.1 Design Documentation

Design Documentation should cover:

1. Functional design specifications;
2. Technical design specifications;
3. A draft version of the Software Test Plan containing at least the test strategy;
4. The Migration Strategy. It should address topics like, data sources, transformations, loading mechanisms, validations, etc..

2.2.1.1. Functional Design Specifications

Functional design specifications will be used as guidelines for the implementation of the system.

They should describe in detail as a minimum of:

- Use Cases representing the system functionalities,
- capabilities and processes,
- interactions with users/systems,
- Traceability matrix between Functional Requirements/Business Rules and Use Cases.
- Technical Design Specifications
- Technical design specifications will be used as a blueprint for the system implementation. They describe how the system will be implemented in order to cope with functional specifications.

They should include as a minimum:

- Conceptual and physical system architecture;
- Software design and layering;
- Modules and components;
- Process, workflows and algorithms design and documentation;
- Interfaces definitions.

2.2.1.2. Draft version of the Software Test Plan

The draft version of the Software Test Plan will serve as the basis for the Software Test Plan to be implemented during "Development and Test" phase. This draft version should include as a minimum:

- Definition of the Software Test Plan Structure and global strategy;
- Reference to the different test phases to be implemented;
- Definition of the test detailed strategy presenting an overall perspective of testing and identifying individual test phase plans for unit, integration, functional, performance, load and stress test phases. Each test phase plan should include at least:
 - Description of the test phase strategy;
 - Test phase standards and practices;
 - Test phase supporting guidelines;
 - Test phase selection criteria;
 - Test phase evaluation metrics;

- Completion criteria for the test phase;
- Test phase implementation templates.
- Reference to the test environment(s) to be used;
- Software Test plan execution planning;
- Software Test team responsibilities and staff.

The final version of the Software Test Plan is to be provided during the Development and Test phase.

2.3. Development and Test

If needed the contractor and/or EMSA may suggest modifying the content of the deliverables of the design phase. These modifications should be agreed by EMSA.

2.3.1 System documentation

2.3.1.1. Operational and Maintenance Documentation

Operational and maintenance documentation must explain how the system should be operated and maintained on a daily base. It should include the following documentation:

- Installation manual;
- Operation and Maintenance manual;
- HOW-TO troubleshooting and root-cause analysis.

2.3.1.2. System building procedures

System building procedures should allow EMSA to completely build the latest version of the system at any moment.

EMSA Build Environment shall be used to automatically execute all system builds procedures. Build procedures shall be used on Maven and/or Ant. Hudson will be the tool used for build automation and Archiva will be the artefacts repository.

At the delivery of the final version the contractor should provide an automatic build procedure with the complete source code, additional software packages and code generators.

For each code generator used during development a correspondent generator should be provided to EMSA.

2.3.1.3. Infrastructure (HW and SW) documentation

The contractor is requested to provide a complete and detailed architecture definition and sizing for the following environment:

- Test;
- Pre-Production/Quality;
- Production.

The environments will be provided at EMSA Data Centre.

In order to correctly size the production environment the contractor must consider the following elements:

- system architecture;
- system implementation;
- non-functional requirements, and
- the performance requirements specified in Annex III - Tender Specifications.

For the production environment, detailed information about requirements for servers' characteristics, network, bandwidth, base software, databases, security and accessibility shall be provided to EMSA. For the others environments, the same level of information must be provided with an indication of expected performance.

2.3.2 Functional Prototypes

During the development phase the contractor is requested to provide intermediary deliveries; three functional prototypes with incremental scope:

- Prototype 1: as a minimum, it demonstrates the visual implementation of the following use cases (e.g. navigation approach, data presentation and clustering; and the ergonomic and design decisions e.g. buttons and data-fields positions and grouping, webpage design solutions, etc.) of the New EMCIP human-machine interface (refer to Appendix D, Requirements and Functional Specifications for more information about the New EMCIP Use Cases):

CMS.CU.03 Incident Notification form;

CMS.CU.06 GUEST Registered;

CMS.CU.07 GUEST Not Registered;

CMS.CU.08 Content visualization;

CMS.CU.09 Restricted EMSA Users;

CMS.CU.10 All EMSA user Contents;

CMS.CU.12 MS user Content;

CMS.CU.13 All MS Content;

EDB.CU.25 View Database Data;

EDB.CU.13 Workflow Management;

EDB.CU.14 Taxonomy Management.

- Prototype 2: Prototype 1 + Core Application Functionalities. Functional Design.

As a minimum, it implements the mandatory (.M) requirements of the Core Application functionalities, it implements the data migration strategy, and it demonstrates the following use cases of the New EMCIP Core Application (refer to Appendix D - Requirements and Functional Specifications):

EDB.CU.03 ReOpen Occurrence;

EDB.CU.04 Accept Occurrence;

EDB.CU.05 Change Investigator in charge;

EDB.CU.06 Lock & UnLock Occurrence;

EDB.CU.07 Reject Occurrence;

EDB.CU.08 Delete Draft Occurrence;

EDB.CU.09 Revoke Acceptance Occurrence;

EDB.CU.10 Edit Occurrence;

EDB.CU.11 Transfer (IMO, Internal or External);

- EDB.CU.12 Submit Occurrence;
- EDB.CU.13 Workflow Management;
- EDB.CU.14 Taxonomy Management;
- EDB.CU.15 Verification rules Management;
- EDB.CU.16 View All occurrences limited;
- EDB.CU.21 Ship Register tool;
- EDB.CU.22 Administrative Module;
- EDB.CU.23 Related Occurrence;
- EDB.CU.24 Automatic Safety Recommendations;
- EDB.CU.25 View Database data;
- EDB.CU.27 Query System;
- EDB.CU.29. Revoke Submission occurrence;
- EDB.CU.30 Delete Submitted occurrence;
- Prototype 3: Prototype 2 + Final Functional Design.
As a minimum, it implements the mandatory (.M) requirements of the Core Application functionalities and it demonstrates the implementation the following use cases of the New EMCIP Core Application (refer to Appendix D - Requirements and Functional Specifications):
 - EDB.CU.01 Investigation Analysis;
 - EDB.CU.02 Data Analysis;
 - EDB.CU.26 GIS Viewer;
 - EDB.CU.28 BI System;
 - OIM.CU.01 Management.
- Enhancement Prototype(s): Prototype 3 + Enhanced Application Functionalities.
It implements the “Nice-to-have” (.N) requirements of the New EMCIP and demonstrates the following use cases of the New EMCIP Enhanced Application (refer to Appendix D - Requirements and Functional Specifications):
 - EDB.CU.01 Investigation Analysis;
 - EDB.CU.10 Edit Occurrence;
 - EDB.CU.14 Taxonomy Management;
 - EDB.CU.16 View All occurrences limited;
 - EDB.CU.26 GIS Viewer;
 - EDB.CU.28 BI System;
 - EDB.CU.29. Revoke Submission occurrence;
 - EDB.CU.30 Delete Submitted occurrence;
 - OIM.CU.01 Management.

For each prototype, the contractor should deliver:

- Applications;
- Functional documentation on functional coverage;
- Draft operational and maintenance documentation;
- System building procedures;
- Source code and supporting documentation;
- Documentation on tests carried out by the contractor on the prototype before its delivery (test specification, test cases, test scripts and test results);
- Release notes;
- Corrective maintenance and operational support, as appropriate.

Please refer to Appendix F – Corrective Maintenance and Operational Support for more details about the corrective maintenance and operational support requirements and processes.

2.3.3 Final Version

The final version of the system delivered to EMSA should contain:

- System implementation;
- Related source codes, build procedures and supporting documentation;
- A complete system documentation;
- Test documentation;
- Release notes;
- Corrective maintenance and operational support, as appropriate.

Please refer to Appendix F – Corrective Maintenance and Operational Support for more details about the corrective maintenance and operational support requirements and processes.

2.3.4 User Documentation

2.3.4.1. User documentation

The user documentation will explain the different components of the system to EMSA's users.

User documentation should include:

- A quick start guide to explain how to access the system and use main functionalities;
- A complete user manual to describe how to use all functionalities of the system;
- On line help content. This will contain a contextual help explaining content and functionalities of every screen of the system and generic help which will provide the same content of the user manual.

Text should be supported by illustrations and screen copies all through the user documentation. User manual and on line help should include a table of content, a glossary and an index.

2.3.4.2. Training support material

As a minimum training support material should contain:

- Training presentations to be displayed during the sessions;
- Practical examples;
- Training data sets - if applicable.

2.3.5 Test Documentation

Test documentation should cover at least 80% of the functionalities of the system.

Tests to be performed by the contractor must cover the two following objectives:

- Verification tests: verify that the product is in line with the functional and technical requirements and design specifications and that implementation best practices were applied,
- Validation tests: verify that procedures and activities as described in the project plan, change management procedures and software development plan were applied.

The test documentation and test results should provide evidence that these objectives are met.

Test documentation should detail all necessary documents to plan, design, execute and report tests. This should include as a minimum:

- The Software Test Plan with all details regarding the test process:
 - Definition of the Software Test Plan Structure and global strategy,
 - Reference to the different test phases to be implemented,
 - Definition of the test detailed strategy presenting an overall perspective of testing and identifying individual test phase plans for unit, integration, functional, performance, load and stress test phases. Each test phase plan should include at least:
 - Description of the test phase strategy;
 - Test phase standards and practices;
 - Test phase supporting guidelines;
 - Test phase selection criteria;
 - Test phase evaluation metrics;
 - Completion criteria for the test phase;
 - Test phase implementation templates.
 - Results achieved with the test phase implementation including at least:
 - Test cases;
 - Test scripts;
 - Data sets;
 - Test results;
 - Test phase report.
 - Reference to the test environment(s) to be used;
 - Software Test plan execution planning;
 - Software Test team responsibilities and staff.

Test results should be added to each test plan once the corresponding tests have been executed.

The contractor will be responsible for preparing all documentation including test cases, test data to be used and test environment.

2.4. Deployment and Data Migration

Deliverables of the deployment phase are:

- Final version of the system deployed and fully working in the three environments
- Updates of the system documentation, user documentation and training materials if needed.
- Updates of the deliverables of the design phase if needed;

- Migration Procedures and Scripts.

2.5. Go-Live

Deliverables of the Go-Live phase are:

- Updates of the system documentation if needed;
- Report on the tasks undertaken by the contractor and their results;
- Final system.

2.6. Training

Training sessions are foreseen for:

- EMSA's operational unit end-users,
- EMSA's IT personnel.

All sessions should mix theoretical and practical parts. In the practical parts, the users actually use the system with hands on examples.

The contractor will be responsible to conduct training sessions. The contractor should provide all supporting material and prepare hands on examples. It will also be responsible for the preparation of the technical environment (software and data).

Training sessions will take place at EMSA's premises in Lisbon. EMSA will provide infrastructure (rooms, IT equipment, video equipment etc.) and will be in charge of administrative organisation of the courses (planning, notifications, and evaluation) and duplication and distribution of course documentation.

3. Timetable

The table below lists the milestones of the project. Dates of delivery and planning will be defined during the initiation phase and recorded in the project plan.

	Dates	Milestones
T0		Signature of the contract
T1	T0+1 week	Kick-off meeting
T2	T0+2 weeks	Prototype 1 - Human Machine I/F Design Acceptance of design documentation
T4	T0+12 weeks	Prototype 1 - End of deployment, acceptance of the final version
T5	T0+14 weeks	Prototype 2 - Core Application. Functional Design Acceptance of design documentation
T6	T0+28 weeks	Prototype 2 - End of deployment, acceptance of the final version
T7	T0+30 weeks	Prototype 3 - Final Functional Design Acceptance of design documentation
T8	T0+44 weeks	Prototype 3 - End of deployment, acceptance of the final version
T9	T0+49 weeks	Data Migration (Design/implementation, testing and acceptance)
T10	T0+11 months	End of training
T11	T0+12 months	Final acceptance of Module 1 – New EMCIP Goes Live
T12	T11+36 months	End of Module 2

The contractor should deliver to EMSA all deliverables of the initiation phase at least 2 working days before the kick-off meeting.

4. Acceptance Procedures

For each deliverable, EMSA provides a formal indication of the acceptance, conditional acceptance or rejection of the deliverable to the contractor.

4.1. Classification of Software Issues

EMSA will classify issues found on software into 3 different categories according to their impact and severity:

- **Blocking issues:** structural problems or serious issues (functional or technical) considered as limitations of the implementation with very high probability of interfering with the expected result. The contractor will be obliged to correct/execute all issues considered in the category,
- **Critical issues:** problems or issues that do not conform to the requirements or specifications or best practices or considered to be the wrong approach to obtain the result, but for each one of them a workaround is available.
- **Minor issues:** changes considered to be a better solution but without a deep impact in the quality of the system. The correction/execution of the issues of this category will be decided case by case.

Each issue is identified and described by EMSA and sent to the contractor. The contractor is requested to track and monitor the treatment of each issue sent by EMSA, using the change management procedures. The acceptance tests and the classification of the issues are made in collaboration between EMSA and the contractor.

The outcome of the acceptance procedure is positive if no issue is found by EMSA. If issues are found by EMSA during the acceptance procedure, the contractor is requested to immediately correct them and the acceptance procedure restarts from the date of the delivery of the corrected deliverable.

EMSA can decide conditionally accept the deliverable when some issues remain uncorrected and that issues are not blocking the system. In order to accept such remaining issues the contractor shall propose a deadline for the correction and EMSA to accept it. The EMSA will take the decision on conditionally acceptance of the product after evaluation of each remaining issue.

No acceptance shall be made by EMSA without a successful execution of the automatic build procedure.

4.2. Documentation

In the case of Project Management documents, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **two weeks** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

In the case System documentation and User Documentation, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **three weeks** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

In the case of Design Documentation, EMSA will provide comment and/or reservations which will be transmitted to the contractor within **three weeks** of the date of delivery. Based on this comment and/or reservations EMSA will either accept or reject the deliverables. In the case of rejection the contractor will be requested to provide a new appropriate revision.

4.3. Final System

The final version of the system will be evaluated by EMSA when available and running on the test, quality and production environments.

Before the final version is accepted EMSA will verify if:

- all issues detected in the previous acceptance tests have been corrected;
- it conforms with the functional specifications;
- it conforms with the technical specifications;
- non-functional requirements are met;
- it works correctly in its environments according to all requirements and specifications.

EMSA will provide issues which will be transmitted to the contractor within **four weeks** of the date of delivery. Based on this issues EMSA will either accept or reject the version. In the case of rejection the contractor will be requested to provide a new appropriate version.

4.4. Final System

The final system will be evaluated by EMSA when the accepted final version will be available in the production environment. EMSA will verify the system operates correctly while being used by end users in real situation.

The final system is accepted within acceptance period (**fourteen days**) at the condition that no blocking issues as described in chapter 4.1 are found.

In the case a blocking issue is found, the acceptance period is frozen until a corrected version is made available on the production environment by the contractor.

4.5. Quality Gate

Delivered source code for Intermediary Deliveries and Final Version shall always be submitted to the Quality Gate defined in Appendix H. Results of the Quality Gate shall be evaluated and decision to accept or reject the release will be taken based on the defined conditions.

5. Meetings

5.1. Project Management Meetings

Action list, risk registry and planning will be reviewed during project management meetings.

At each project management meeting, the contractor should present an updated project status report.

In addition to the project status reports, between the project management meetings, the contractor delivers to EMSA a flash report.

The contractor is responsible for providing detailed agenda and supporting documents for the meetings, support the discussions during the meeting, and providing the minutes of the meetings. The detailed agenda and supporting documents must be provided by the contractor three days before each meetings. The minutes of the meetings must include at least the topics discussed, decisions taken and action items with indication of the responsible person and deadline of the actions.

[End of document]

European Maritime Safety Agency

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

