SUPPLIER DOCUMENT Cover Page

idm@F4E UID / VERSION
2R5FL3 / 1.1

VERSION CREATED ON / STATUS
04 January 2017 / Approved

EXTERNAL REFERENCE

Supplier Document

D4 MARTe Configuration Management Plan

Under the Specific Contract no. F4E-OFC-361-06, with subject “Fast plant controller prototype”, this document is deliverable D4 MARTe Configuration Management Plan (CMP).

The scope of this document applies to the software development activities of the MARTe project. It should be considered as guidance for configuration management activities in the framework of this project.

Approval Process			
	Name	*Action*	*Affiliation*
Author	**Herrero I.**	**04 January 2017:signed**	
Co-Authors			
Reviewers			
Approver	**Cabrita Neto A.**	**06-Jan-2017: approved**	**ITERD**
RO: Cabrita Neto Andre (F4E)			
Read Access	**LG: GTD team, LG: OFC-361-06-CCFE, AD: IDM_F4E, AD: F4E-A40_HEAD, AD: I-CODAC, AD: IDM IE-TS-CO-00 CODAC, GG: IAC, GG: IAS Audit on Document Management, project administrator, RO**		

Orig. Document MD5#: D1B1BCB63D5F349A1067FBE3EE52B33C

Change Log			
D4 MARTe Configuration Management Plan (2R5FL3)			
Version	***Latest Status***	***Issue Date***	***Description of Change***
v0.0	In Work	17 July 2015	
v1.0	Approved	17 July 2015	Document creation.
v1.1	Approved	04 January 2017	Clarifications on branch types. Add explanation for repositories.

Identification of the document			
Document Reference	01.ES-F4E-OFC-361-06 D4	Revision	1.1
F4E Reference	F4E_D_2R5FL3	F4E TRO	André C. Neto
F4E Customer Reference	N/A		
Date	2017-01-04		
Supplier	GTD SISTEMAS DE INFORMACION		
Graded Quality level	Class 3 – Any safety related item (SR) or non safety related item (NSR) whose failure could result in MODERATE impact.		

MARTe Configuration Management Plan

Fast plant controller prototype (F4E-OFC-361-06)

Summary

Under the Specific Contract no. F4E-OFC-361-06, with subject “Fast plant controller prototype”, this document is deliverable MARTe Configuration Management Plan (CMP).

The scope of this document applies to the software development activities of the MARTe project. It should be considered as guidance for configuration management activities in the framework of this project.

	Written by	Revised by	Approved by
Name	Ivan Herrero	Ivan Herrero	Javier Varas
Signature			
Date	2017-01-04	2017-01-04	2017-01-04

Contact person GTD: javier.varas@gtd.eu
Phone: +34 93 493 93 00

DISSEMINATION

Distributed to	Copies	Means
Project Team	1	e-mail
Fusion for Energy	1	Electronic

SUMMARY OF MODIFICATIONS

Edition	Date	Chapter	Modification	Author/s
1.0	17/07/2015	All	Document creation	IH
1.1	04/01/2017	2.4 Change Control 2.8 Repositories	Clarifications on branch types. Add explanation for repositories.	IH

CONTENTS

INDEX OF FIGURES

INDEX OF TABLES

0 INTRODUCTION

0.1 Purpose and scope of the document

Under the Specific Contract no. F4E-OFC-361-06, with subject “Fast plant controller prototype”, this document is deliverable MARTe Configuration Management Plan (CMP).

The scope of this document applies to the software development activities of the MARTe project. It should be considered as guidance for configuration management activities in the framework of this project.

0.2 Relationship to other plans

Next figure shows the relationship between this plan and the other plans referenced in this document.

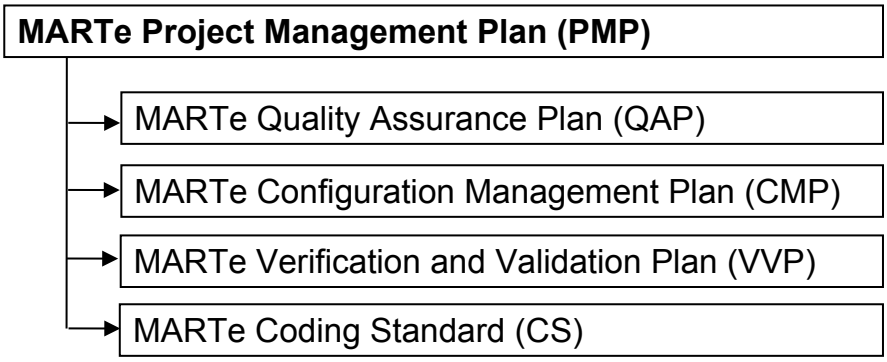

Figure 1 Relationship to other plans

Quality, configuration management and verification processes are known as integral processes. That is, they are present during the entire software lifecycle of MARTe.

0.3 MARTe framework overview

MARTe (Multithreaded Application Real-Time executor) is a multi-platform C++ real-time control middleware, with a simulink-like way of describing the problem. It allows for a modular development and execution environment to control systems, which ensures and monitors real-time, and facilitates test and commissioning.

The project objective is to develop a MARTe2 version, which will be the result of a reduction exercise of the core framework with the lessons learned from MARTe. This version will incorporate an integral quality assurance process.

0.4 Applicable and reference documents

Ref.	Code	IDM / Date	Ver.	Description
AD1	MARTe-P-PMP-01	F4E_D_2RBL9F	1.0	MARTe Project Management Plan

Table 1 Applicable documents

Ref.	Code	IDM / Date	Ver.	Description
RD1	MARTe-P-CIE-01	F4E_D_2JEP9G	1.2	MARTe Continuous Integration Environment
RD2	MARTe-P-DEN-01	F4E_D_2SB7R8	1.1	MARTe Development Environment
RD3	MARTe-P-VVP-01	F4E_D_3S94CT	1.0	MARTe Verification and Validation Management Plan
RD4	MARTe-P-QAP-01	F4E_D_2F888J	1.0	MARTe Quality Assurance Plan

Table 2 Reference documents

0.5 Definitions, acronyms and abbreviations

Term	Definition
AD	Applicable Document
API	Application Program Interface
CCS	CODAC Core System
CM	Configuration Management
ITER	International Thermonuclear Experimental Reactor
F4E	Fusion for Energy
IDM	ITER Document Management
MARTe	Multithreaded Application Real-Time executor
RD	Reference Document

Table 3 Definitions, acronyms and abbreviations

1 ORGANIZATION AND ENVIRONMENT

1.1 Organizational Responsibilities and Interfaces

The MARTe project manager is responsible for the configuration management process. He shall be assisted by selected F4E team for some tasks. According to this principle, F4E shall ensure that all the objectives required by this plan in terms of configuration management for software lifecycle artefacts are fully satisfied.

The MARTe project manager shall be the key person in charge of the configuration management process and shall play the interface role in that sense with the Brainstorming Group [AD1§2.1] and/or the rest of the project team.

1.2 Tools and Methods

Concerning configuration management activities, the following tools will be used in the framework of the project:

Git and GitLab

GIT and GITLAB are the tools for source control version. They offer a large set of features that allow to efficiently and safely distributing source code. In particular they are widely used and distributed, besides being currently used to manage the sources of some of the most prestigious systems in the world.

Please refer to documents [RD1] and [RD2] to setup the continuous integration server and to use the development environment accordingly to this CMP.

2 ACTIVITIES

2.1 Configuration Identification

The following table summarizes which data shall be identified as configuration items during the entire software lifecycle, indicating as well the time points of the project where these items shall be developed and updated.

Item	Identifier	Rel.1	Rel.2	Rel.3	Rel.4
MARTe Project Management Plan	MARTe-P-PMP-01	●			
MARTe Architecture Design Document	MARTe-P-ADD-01	●	↻	↻	↻
MARTe Coding Standard	MARTe-P-CS-01	●			
MARTe Quality Assurance Plan	MARTe-P-QAP-01	●			
MARTe Configuration Management Plan	MARTe-P-CMP-01	●			
MARTe Verification and Validation Management Plan	MARTe-P-VVP-01	●			
MARTe Tutorials	MARTe-P-TUT-01	●	↻	↻	↻
MARTe Development Environment	MARTe-P-DEN-01	●			
MARTe Continuous Integration Environment	MARTe-P-CIE-01	●			
MARTe Portability	MARTe-P-POR-01	●			
MARTe Software Requirements Specification	MARTe-P-SRS-01	●	↻	↻	↻
MARTe Source Code	MARTe_SRC	●	↻	↻	↻
MARTe Testing files	MARTe_TST	●	↻	↻	↻
MARTe Quality Assurance Records	MARTe_QAR_XXX	●	↻	↻	↻

Table 4 Configuration identification of data items (●: created ; ↻: updated)

Identifiers of data items marked as XXX (Problem Reports, Quality Assurance Records, Configuration Management Records and Change Requests) shall be managed via redmine [RD3].

2.2 Baselines and Traceability

Major baselines shall be established during the project development at the light of the results of the main reviews planned during the software lifecycle. These major baselines shall be established once the defined transition criteria between specified processes will be accomplished.

The following table summarizes the major baselines defined during the entire project, indicating as well, the establishment condition associated to the baselines and the expected time (in relationship with the expected reviews) when these baselines shall be set up.

Baseline Identifier	Description	Establishment Condition	Expected Time Point
Rel.1	Reduced version	All core functionalities passing QA process.	December 2015
Rel.2	Integration in CCS	MARTe2 ready to interoperate with CCS.	March 2016
Rel.3	Deployment in testing rig.	MARTe2 deployed in F4E-OFC-361-07-01 testing rig.	July 2016
Rel.4	Final version	MARTe2 ready for real-world applications.	January 2017

Table 5 List of major baselines

Additionally, a set of minor baselines shall be defined during the project in order to freeze artefacts associated to a certain moments of time (for example, following some Sprints).

As reflected above, the proposed baseline definition approach is oriented to planned reviews. No baselines shall be established according to the functional behaviour of the code.

2.3 Problem Reporting

The objective of problem reporting, tracking and corrective action is to record process non-compliance with software plans and standards, to record deficiencies of outputs of software lifecycle processes, to record anomalous behaviour of software products, and to ensure the resolution of these problems.

A problem can be detected either internally (F4E/GTD) or externally (Brainstorming Group/others) to the development team, through casual observation, review, audit or formal test.

Once detected, the problem reporting process shall follow the guidelines reflected in [RD4]. Problem reports will be reflected as redmine issues.

2.4 Change Control

The objective of the change control activity is to provide for recording, evaluation, resolution and approval of changes throughout the software lifecycle.

All the configuration identified artefacts shall be stored in git repositories. The project team will get read access rights to all the branches of the repositories, but write access will be restricted upon the role of the member.

Unplanned changes upon configuration identified artefacts shall be submitted to a change request or problem reporting processes.

Once identified, the change request process shall follow the guidelines established in [RD4].

2.4.1 Git workflow for MARTe

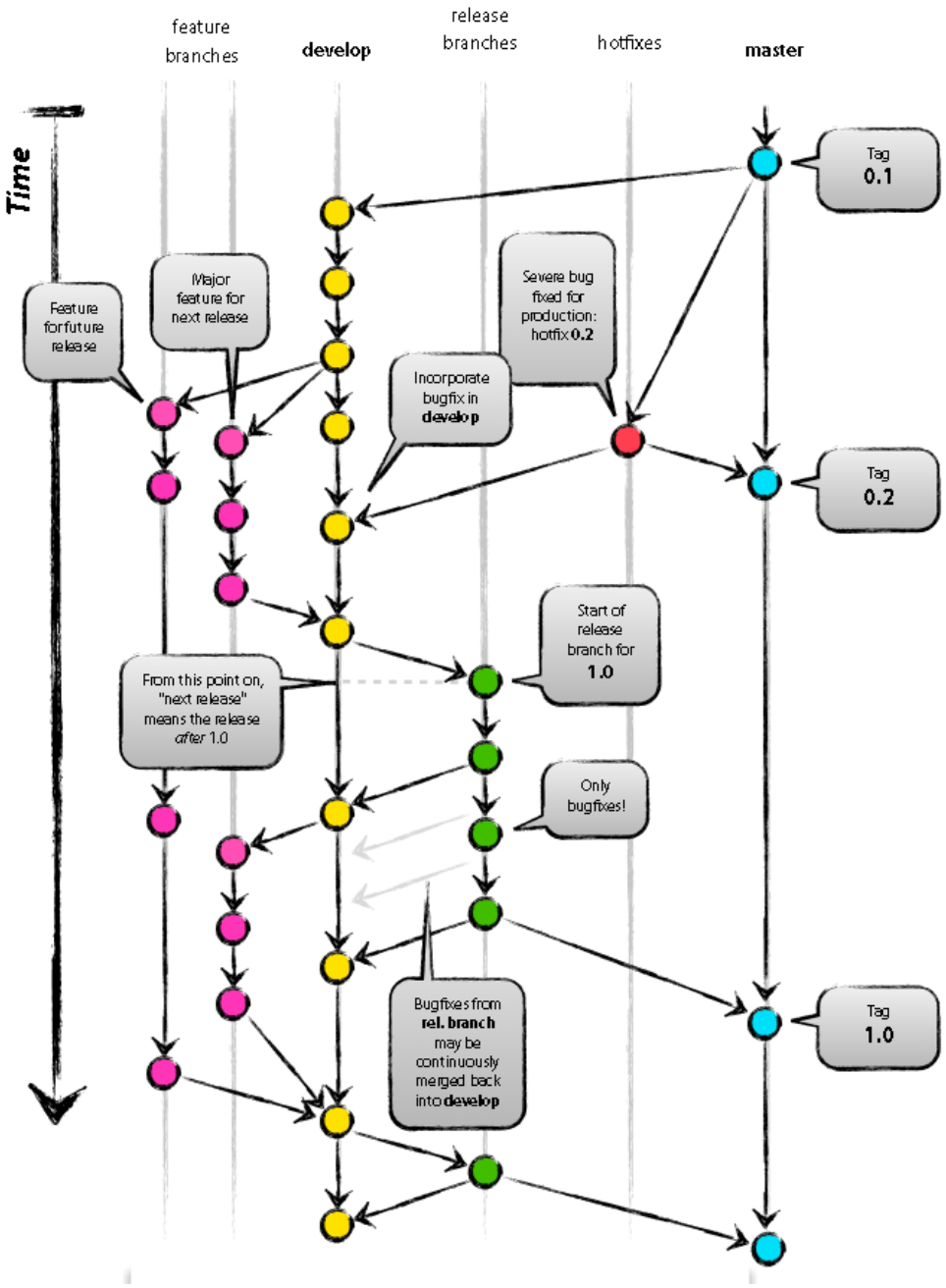

Figure 2 Git workflow

Master branch:

- Main branch. It reflects a production-ready state.
- Only MARTe project manager will push to this branch, so he will be the only with write access rights.
- Each production-ready version will be tagged following the convention “vX.Y” (e.g. v0.1, v0.2, v.015, v1.0, v1.1 …).

Develop branch:

- Latest delivered development changes for the next release.

- All members of the developing team are allowed to push at this branch, so all of them have write access rights.

Feature branches:

- New features for the upcoming releases.
- Each member of the developer team will create features branches locally on its private repository.
- Merged back into develop branch when the features are ready for the next release.
- After merged into develop branch, the feature branch shall be deleted.
- Members of the developer team are encouraged to push their feature branches to the central repository and to regularly push new commits, as a means to have an informal backup. However, it is mandatory to push a feature branch to the central repository when a peer review is going to be run through by a team mate.

Release branches:

- Support preparation of new production releases.
- Minor bug fixes and preparing meta-data for a release (version number, build dates ...).
- Adding large new features is prohibited.
- At the start of the branch, the upcoming release gets assigned a version number.
- Merged into master branch when they are ready to become a real release and into develop branch, so that future releases also contain the bug fixes.
- These release branches will be created and maintained by the MARTe project manager, who shall delete them after merged on master.

Hotfix branches:

- Used to resolve critical bugs in a production version.
- When resolved, the branch will be merged back into master and develop branches.
- When a release branch exists, the hotfix changes need to be merged into that release branch, instead of the develop branch.
- These hotfix branches will be created and maintained by the MARTe project manager, who shall delete them after merged on master.

2.5 Change Review

MARTe change review procedure reflects the method of handling feedback from and to the software lifecycle processes; the methods of assessing and prioritizing problems, approving changes, and handling their resolution or change implementation; and the relationship of these methods to the problem reporting and change control activities.

The workflow to change any identified configuration item as a result of detecting a problem and/or requesting a change shall be compliant with Problem Reporting and Change Request procedures explained in sections §2.3 and §2.4 respectively.

2.6 Configuration Status Accounting

The objective of the status accounting activity is to provide data for the configuration management of software lifecycle processes with respect to configuration identification, baselines, problem reports, and change control.

To comply with the above mentioned objective, the following activities/resources shall be activated:

- GIT server shall be accessible for the project team through the execution of GIT clients. Using those GIT clients, users could get information about defined baselines in the source coded files.

2.7 Archive, Retrieval and Release

The objective of the archive and retrieval activity is to ensure that the software lifecycle data associated to the software product can be retrieved in case of need to duplicate, regenerate, retest or modify the software product. The object of the release activity is to ensure that only authorized software is used, especially for software manufacturing, in addition to being archived and retrievable.

2.7.1 Archive and Retrieval Process

F4E shall be capable of retrieving software lifecycle data associated with the software product from the files' data server which is mounted in F4E's facilities.

Integrity of the stored data shall be ensured through the following methods:

- Only the members of the development team shall get the access rights needed to modify something in previously configuration managed artefacts.
- Data items will be physically archived in high quality hard disks, managed by F4E-IT.
- Backup copies shall be performed on data servers to ensure regeneration if needed. This will be assured by F4E-IT.

Retrieval will be also assured by F4E-IT.

2.7.2 Software and Document Release

One release shall be prepared for each established baseline (see Section §2.2). Preparation and package of the releases will be under F4E responsibility. Each release will include all artefacts (software and documents) associated to the correspondent baseline (see Section §2.2).

Each release shall be associated to a delivery note. This document (configuration management record) shall be the evidence associated to the delivery of a specific release.

2.8 Repositories

The CM activities are meant for the development of the MARTe2 framework, which is expected to be managed into its own git repository. Nevertheless, standard components based on MARTe2 will be developed on an independent git repository, following all the rules applied to the MARTe2 framework development itself. This repository of MARTe2 components will need a link to the MARTe2 repository, in order to be compiled and tested.

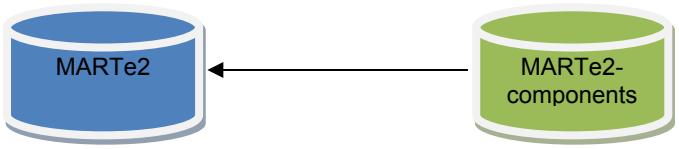

Figure 3 Repositories

3 TRANSITION CRITERIA

This section describes the transition criteria for entering the project configuration management (CM) process.

CM is intended as an integral process. That is, this is an active process from the beginning of the software lifecycle until the end.

The project team, supervised by MARTe project manager, as defined in previous sections, shall be in charge of introducing artefacts into configuration management. They should consider the maturity of the artefact before doing it. Otherwise, it could result in an excessive number of iterations for verification activities.

4 CM ARTEFACTS

This section defines the software lifecycle artefacts produced by the CM process, including:

- Documentation: made up of templates, configuration files for different tools, and models.
- Source code: all the *.cpp, *.h and makefiles
- Testing code: all the *.cpp, *.h and makefiles
- Reports: mainly redmine PDF records resulting from V&V activities (refer to [RD3])

Other type of records such as change history records will be available upon request.

5 CONTRIBUTORS CONTROL

As defined previously, CM process shall be entirely under F4E's responsibility. Thus, no specific CM process requirement to control external contributors is applied at this point in MARTe development roadmap.