Embedded World

T-EMB-800

Faillist

Request for proposal

Table of contents

Introduction

The aim of this request of proposal (RFP) is to present the requirements for the Faillist project.

1. Company description

EpiPlanes is a French multinational company which Is an Airbus subcontractor.

Founded in 2000 in the dynamic city of Nantes, near the production factory located in Bouguenais, the society turnover is continually increasing since its creation.

EpiPlanes has been named first supplier for Airbus' new project. The company needs to supply a failure reporter software which will be integrated in the next plane, currently developed by Airbus.

2. Project context

The provider will have to create a failure reporter software. The aim of this software is to write in a log failure file all error codes raised by the plane. To dynamically test the extraction, the supplier will have to supply a test application. The software must follow the DO-178B norm and the level must be DAL A, except the test application. This implies that unit test with DAL A level must be created.

From a documentary point of view, the provider will have to supply all necessary documents that will improve independency, traceability and verification processes.

3. Requirements

The software must implement the following requirements and are organized in four categories:

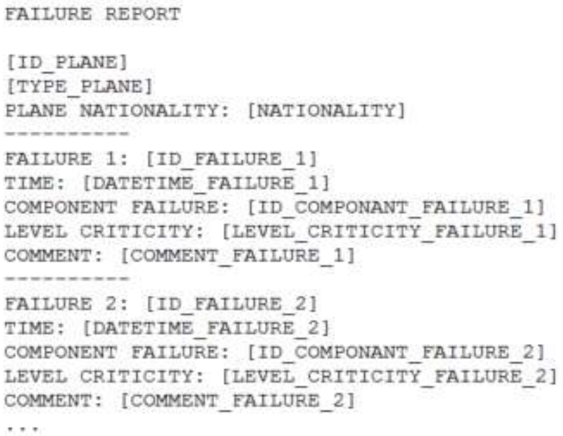
- REQ_DESIGN_XXX: design requirements,
- REQ_FUNC_XXX: functional requirements,
- REQ_TEST_XXX: Test requirements,
- REQ_DATA_XXX: data managing requirements.

a. Design requirements

REQ_DESIGN_010	The application must be developed and test following DO-178B DAL A norm.
REQ_DESIGN_020	Each file .c and .h must contain only one piece of code.
REQ_DESIGN_030	Each function must be unit tested, except TestSettings application.
REQ_DESIGN_040	Dynamic allocations are forbidden.
REQ_DESIGN_050	Each defined value or type must be commented.
REQ_DESIGN_060	Each function prototype must be commented using doxygen. It must contain detail on: • Prototype • Requirement(s) id's • Param(s) • Return value • Any details that you judge useful.
REQ_DESIGN_070	Each C function must be commented.

b. Functional requirements

REQ_FUNC_010	Raised errors must be encoded with the following format:
	{
	Id_plane: ID PLANE,
	Type_plane: TYPE PLANE,
	Nb_failures: NB FAILURES,
	failures: [
	{
	Id_failure: ID FAILURE,
	Date: DATETIME FAILURE,
	Id_component: ID COMPONENT FAILURE,
	Level_criticity: LEVEL CRITICITY LEVEL,
	Comment_failure_size: COMMENT FAILURE SIZE,
	Comment_failure: COMMENT FAILURE
	},
	]
	}

REQ_FUNC_020	Field ID_PLANE must contain the identification of the plane.
REQ_FUNC_030	Field TYPE_PLANE must contain the type of the plane.
REQ_FUNC_040	Field NB_FAILURES must indicate the number of raised failure.
REQ_FUNC_050	Field DATETIME_FAILURE_X must indicate the date and the time of the error X.
REQ_FUNC_060	Field ID_COMPONENT_FAILURE_X must indicate which component is concerned by the error X.
REQ_FUNC_070	Field LEVEL_CRITICITY_FAILURE_X must indicate which is the criticity level of the failure X.
REQ_FUNC_080	Field ID_FAILURE_X must indicate the id of the failure X.
REQ_FUNC_090	Field COMMENT_FAILURE_X_SIZE must indicate the size in bytes of the CAUSE_FAILURE_X field.
REQ_FUNC_100	Field COMMENT_FAILURE_X must describe the failure X cause.
REQ_FUNC_110	The file must contain data extracted from the error code described in REQ_FUNC_010.
REQ_FUNC_120	The file must have the following contents:  FAILURE REPORT [ID_PLANE] [TYPE_PLANE] PLANE NATIONALITY: [NATIONALITY] ---------- FAILURE 1: [ID_FAILURE_1] TIME: [DATETIME_FAILURE_1] COMPONENT FAILURE: [ID_COMPONANT_FAILURE_1] LEVEL CRITICITY: [LEVEL_CRITICITY_FAILURE_1] COMMENT: [COMMENT_FAILURE_1] ---------- FAILURE 2: [ID_FAILURE_2] TIME: [DATETIME_FAILURE_2] COMPONENT FAILURE: [ID_COMPONANT_FAILURE_2] LEVEL CRITICITY: [LEVEL_CRITICITY_FAILURE_2] COMMENT: [COMMENT_FAILURE_2] ...
REQ_FUNC_130	Field [TYPE_PLANE] in REQ_FUNC_120 must be converted to a readable type string which can be retrieved in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “TYPE_PLANE”, column “Type plane”.

REQ_FUNC_140	Field [NATIONALITY] in REQ_FUNC_120 must be retrieved from the [ID_PLANE] field (see “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_PLANE” sheet for more precisions).
REQ_FUNC_150	Field [ID_FAILURE_X] in REQ_FUNC_120 must be converted to a readable id string which can be retrieved in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_FAILURE_X”, column “Failure type”.
REQ_FUNC_160	Field [DATETIME_FAILURE_X] in REQ_FUNC_120 must be converted to a readable date following the format “YYYY/MM/DD – hh:mm:ss” with: • YYYY: the year of the error • MM: the month of the error DD: the day of the error • hh: the hour of the error • mm: the minute of the error ss: the second of the error
REQ_FUNC_170	Field [ID_COMPONENT_X] in REQ_FUNC_120 must be converted to a readable id string which can be retrieved in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_COMPONENT_FAILURE_X”, column “Component”.
REQ_FUNC_180	The name of the extracted file must follow this format: Extraction_report_[ID_PLANE]_[DATETIME].txt
REQ_FUNC_190	Field [ID_PLANE] in the name of the extracted file must be the ID_PLANE contained in the error code.
REQ_FUNC_200	Field [DATETIME] must be the date of the extraction.

c. Test requirements

REQ_TEST_010	A testing application must be done. Its name is TestSettings
REQ_TEST_020	The design of TestSettings is free.
REQ_TEST_030	TestSettings must generate an error code, display it in a terminal and launch the Failist application.
REQ_TEST_040	Two types of error code generation are possible: manually and automatically.
REQ_TEST_050	Manually generation: user can create a custom error code of 1, 2 or 3 failures. All fields can be set by the user.
REQ_TEST_060	Automatically generation: user choose the number of failures and TestSettings generates them randomly.
REQ_TEST_070	Each function must be unit tested, except those of TestSettings application.
REQ_TEST_080	Each test case in an unit test must be described.
REQ_TEST_090	Statement coverage must be fulfilled.
REQ_TEST_100	Decision/Condition coverage must be fulfilled.
REQ_TEST_110	MCDC (modified condition decision coverage) must be fulfilled.
REQ_TEST_120	Independency must be satisfied.
REQ_TEST_130	Each developer must write at least one unit test.
REQ_TEST_140	Each test must be reviewed.
REQ_TEST_150	3 persons must perform in the development of one piece of code: • The developer: which writes the piece of code • The tester: which tests the piece of code • The reviewer: which ensures that the test is correctly written

d. Data requirements

REQ_DATA_010	Field ID_PLANE is an array of unsigned char[12].
REQ_DATA_020	The description of possible values of the field ID_PLANE is described in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_PLANE”.
REQ_DATA_030	ID_PLANE string must be terminated by a ‘\0’.
REQ_DATA_040	Field TYPE_PLANE is an unsigned int value.
REQ_DATA_050	Field TYPE_PLANE must be one of the values described in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “TYPE_PLANE”.
REQ_DATA_060	Field NB_FAILURES is an unsigned int value.
REQ_DATA_070	Field NB_FAILURES range is [0;999].
REQ_DATA_080	Field DATETIME_FAILURE_X is an unsigned int value
REQ_DATA_090	The date and time format of the DATETIME_FAILURE_X field is the number of second since 00:00 January the 1rst 1970 UTC.
REQ_DATA_100	Field ID_FAILURE_X is an unsigned int value.
REQ_DATA_110	Field ID_FAILURE_X must be one of the values described in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_FAILURE_X”.
REQ_DATA_120	Field ID_COMPONENT_FAILURE_X is an unsigned int value.
REQ_DATA_130	Field ID_COMPONENT_FAILURE_X must be one of the values described in file “Embedded_World-Request_for_proposal-Annex1.xlsx”, sheet “ID_COMPONENT_FAILURE_X”.
REQ_DATA_140	Field LEVEL_CRITICITY_FAILURE_X is an unsigned int value.
REQ_DATA_150	Field LEVEL_CRITICITY_FAILURE_X range must be [0;10], with 0 the minimum level of criticity, 10 the maximum.
REQ_DATA_160	Field COMMENT_FAILURE_X_SIZE is an unsigned int value.
REQ_DATA_170	Field COMMENT_FAILURE_X_SIZE range is [0;MAX_COMMENT_SIZE].
REQ_DATA_180	MAX_COMMENT_SIZE is equal to 1000.
REQ_DATA_190	Field COMMENT_FAILURE_X is an array of unsigned char[MAX_COMMENT_SIZE].
REQ_DATA_200	Field DATETIME of the file name must follow the format: YYYY_MM_DD_hh_mm_ss with: • YYYY: the year of the extraction • MM: the month of the extraction • DD: the day of the extraction • hh: the hour of the extraction • mm: the minute of the extraction ss: the second of the extraction

4. Delivery conditions

Repository name	T-800-EMB-Faillist_$ScolarYear
Repository right	ramassage-tek
Language	C
Platform target	Embedded
Project sources	./
Tests sources	./tests/

The delivery package shall contain the following documents:

- Documentary:
  - SAS (Software Architecture Specifications). This document must follow the template “Template – Software Architecture Specifications.docx” given by the company. It must contain an explanation of the provider understanding of the project, constraints and solutions proposed, global and detailed UML diagrams of the architecture proposition, and the state machine diagrams if needed.
  - SQS (Software Qualification Specifications). This document must follow the template ‘Template – Software Qualification Specifications.docx” given by the company. It must contain tests procedures to check the software. The document must also contain a traceability matrix which link the requirement id with the test id. The provider should write a first version of this document only from this request for proposal and before any work.
  - SQSA (Software Qualification Specifications Acceptance). This is the SQS document filled with results of the tests procedures.
  - Reviews of unit tests. “Template-Review.xlsx”
  - Truth Table associated to each unit test – “Template-TruthTable.xlsx”
  - Doxygen
- Tests:
  - Unit tests written with NovaProva

5. Planning

The planning of this project consists of several deadlines, T0 is the beginning of the project:

Kick-Off	T0
Bootstrap	T0
Follow up	T0 + 2 weeks
Follow up	T0 + 4 weeks
Delivery	T0 + 4 weeks
Review	T0 + 5weeks