**UNIVERSIDAD TECNOLÓGICA DE QUERÉTARO**

**CESEQ**



**Diplomado en Software Embebido**

Proyecto Integrador

<File Name: Software Development Plan.docx/etc>

DOCUMENT: Software Development

Document No. #CESEQ\_SDP\_001

Scrum Master: Ramírez, Jesus

Developer. Sánchez, Luis

Date (YYYYMMDD): 20190405

Version: 1.0.0.

Project Version: x.x.x.

# Log

Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.0.0. | 20190405 | First release | Pérez, Adbeel |
|  |  |  |  |

Project Document Version

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date (yyyymmdd) | Description | Reviewer |
| 1.0.0. | 20190823 | Initial Release | A: Luis Sánchez  R: Jesús Ramírez |
|  |  |  |  |

# Index

Table of Contents

[1. Log 2](#_Toc18056543)

[2. Index 3](#_Toc18056544)

[Table of Figures 3](#_Toc18056545)

[3. Project Scope 4](#_Toc18056546)

[Software 4](#_Toc18056547)

[Documentation 4](#_Toc18056548)

[System requirements 5](#_Toc18056549)

[4. Deliverables 5](#_Toc18056550)

[5. Development methodology 6](#_Toc18056551)

[6. Estimates 6](#_Toc18056552)

[7. Planning 7](#_Toc18056553)

[8. Solving Problem Strategy 8](#_Toc18056554)

[9. Design 8](#_Toc18056555)

[9.1. Standards 8](#_Toc18056556)

[9.2. Naming conventions 8](#_Toc18056557)

[10. Testing 9](#_Toc18056558)

[10.1. Verification strategy (black box test) 9](#_Toc18056559)

[10.2. White box strategy 9](#_Toc18056560)

[10.3. Cyclomatic Complexity Redundance index 9](#_Toc18056561)

[11. Release 10](#_Toc18056562)

[11.1. Software Development Folder 10](#_Toc18056563)

[11.2. Integration Tests Strategy 10](#_Toc18056564)

[11.3. Validation Testing / Functional Testing 10](#_Toc18056565)

[11.4. Throughput and Flash and RAM measurement 11](#_Toc18056566)

[12. Results 11](#_Toc18056567)

[13. Lessons Learned 11](#_Toc18056568)

# Table of Figures

Figure 1 Project folder structure. 4

# Project Scope

The Project: Proyecto Integrador, is an embedded solution which has software, hardware and, documents artifacts.

### Software

The software portion are the code files which generates the application. The application is a synergy project using e2 studio GUI. The source code is written in C/C++ language and consists in s .C and .H files.

The project also includes files and functions to configure ports, interruptions, and registers. Besides, the synergy project includes files for configuration, scripts, debug information, and other auxiliary files. For this reason, it is included a repository folder as part of the project folder structure.

### Documentation

The documentation is located in the project folder structure (Figure 1). It has the material and references to support the solution. It has been uploaded to the following Git Hub repository:

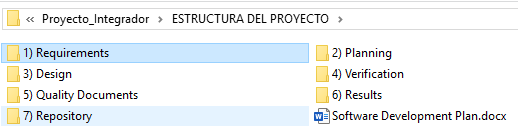
<https://github.com/Luis1250/Proyecto_Integrador>

Figure Project folder structure.

#### Project folder structure

The repository has the following folder structure

* Requirements: At this point there is not any deviation. So, the customer requirements described in the document *20190527 Requisitos del proyecto integrador CESEQ.pdf* located in the repository under de folder: *1) Requirements\stakeholder* **is fully accepted**.
* Planning: All the planning documentation is located in the repository under the folder: *2) Planning:*
  + The file *Planning.docx* includes a Gantt with the tasks and they time window for completion.
* Design: The design artifacts are located in the repository under the folder *3) Design*
  + The design includes UML diagrams to define the classes and modules of the project.
* Verification: The verification strategy is located in the repository under del folder *4) Verification*
  + The verification consists in a document with the verification strategy to demonstrate that the software has quality.
* Quality Documents: The quality documents are located in the repository under the folder: *5) Quality Documents*
  + The quality documents includes the standards to develop the software.
* Results: The results of the project are located in the repository under the folder: *6) Results*
* Repository: Contains the software solution for the project.
  + The results includes material to demonstrate that the project followed the plan, design, and any other artifact mentioned in the documentation.
* Repository: The repository contains the working area where all the contributors can deliver code to complete the project.

#### Stakeholder requirements

The stakeholder requirements are the inputs of this project, they contain the critical aspects that the client is expecting to be included in the final solution.

The document is located in the following path of the repository:

ESTRUCTURA DEL PROYECTO\1) Requirements\stakeholder

## System requirements

The system team has decided that the solution is not big enough to have system requirements, high level requirements (subsystem) and low-level requirements. The decision is to have just software requirements. These software requirements have the enough level detail to be implemented by the development team. Furthermore, the software requirements are considering the testing to cover completeness and correctness of the project.

The software requirements are located in the Software Requirements Specifications documents located in the following path of the repository:

ESTRUCTURA DEL PROYECTO\1) Requirements\SRS.docx

This section **MUST** be contained in this document or in a different document indicating the path in this section, in case a new document needs to be created then it **SHALL** be contained at:

<PROJECT\_PATH>\ 1) Requirements\3. SWRA\_20190405.xlsx

# Deliverables

Since this Project is academic and the team wants to probe that the process has been followed, the following folders in the repository are considered inputs:

* + ESTRUCTURA DEL PROYECTO\1) Requirements
  + ESTRUCTURA DEL PROYECTO\2) Planning
  + ESTRUCTURA DEL PROYECTO\3) Design

The outputs are:

* + ESTRUCTURA DEL PROYECTO\4) Verification
  + ESTRUCTURA DEL PROYECTO\5) Quality Documents
  + ESTRUCTURA DEL PROYECTO\6) Results

Basically, the inputs define the final product. The requirements folder contains the customer requirements and the team’s requirements. These inputs allow to design a solution but this design is just another input for the development team. That is why the design is considered an input.

The verification folder has the evidence that the software implements the specification described in the requirements and design. The document containing the evidence and the strategy to get that evidence is called Software Verification Plan (*SoftwareVerificationPlan.docx*)

Quality Documents folder is considered other output since it has the evidence that the project followed all the expectations described in the *Lista de Cotejo de Auditoria de calidad V1.1.docx*

The results folder collects all the outputs which are not related with the other output folders. It includes but not limited to: evidence of meetings, design drafts, preliminary results, estimations, etc.

## Software delivering process

The delivering process consist in three phases:

1. Developmental phase: This phase is for the development team. This phase is the longer in time because during this phase the development team can deliver at any time code updates without restrictions. The development team is delivering in a branch to correspond with the functionality to complete.
2. Integration phase: This phase occurs after a developmental phase. The duration of this phase takes at much one day. The objective is to integrate the functionality worked during the developmental phase. So, the branch is merge with the master branch and a new branch is created to continue working with the next functionality.
3. The Final Release: This is a branch and the intention of this branch is to be last one worked. So, this is also the last integration with the master branch and it is reserved to the completion of the project. During this phase, the last development phase is completed (release 4), the final release branch is created and any pending task will be work here. During the final release, the quality team can request modifications to any artifacts in order to be corrected. The testing team will also check the correctness and completeness of the verification process.

Include in this section the delivering process and dates if it applies

All the artifacts worked and presented as results will be taken from the repository. The last state of the results will be the master branch after the Final Release branch has been merged with the master. This stage impacts to all the documentation and the e2 studio working area.

# Development methodology

The team has decided to use an agile methodology to accelerate the development process.

Scrum is the agile methodology selected and has the following controls:

* Scrum board:
  + Product Owner: Luis Sánchez
  + Scrum Master: Jesús Ramírez
  + Development team:
    - Luis Sánchez
    - Jesús Ramírez
* Sprint length
  + Each sprint is scheduled in one week in order to be synchronized with the releases.
* Planning meetings
  + Are scheduled at the beginning of each release in order to plan the tasks for the next release.
  + The retrospective meeting will be done after the closure of the release and before the next planning in order to apply any improvement proposed during the retrospective meeting.
  + All the team members SHALL be required to attend all the meetings.

# Estimates

* Estimates **SHALL** contain all the Inputs, like:
  + Hw Facts:
    - Board availabilty
    - Plant availability
    - PC availability
    - Osciloscope
    - Signal generator
    - Multimeter
    - Power supply
  + Activities Facts
    - Human resources
      * Product owner should be the tutor
      * Master scrum and developer (better called as leader)
      * Number of Developers
  + SW Facts
    - Operative system form Renesas works.
  + Hw Assumptions:
    - Hardware damaged.
    - Laboratory time availability.
  + Activtities Assumptions
    - Team time availability.
    - Hardware in good conditions.
  + SW Assumptions
    - Programming language, SW IDE or Hw platform unkown.
    - SW Module unknown.
* It **SHALL** be defined all risks, remember that this is an input for the FMEA:
  + UTEQ holidays.
  + Team is not complete due other projects or trips.
  + New hardware and microcontroller
  + Error in the OS Configuration.
  + Error hardware connection.
* It **SHALL** have a breakdown of all task and activities that are needed and analyzed their dependency between them, some good examples to estimate are:
  + **Activities etimated**
    - **Create and update documents** (design planning verification and so on). Consider the time to create and update documents (SDP, schedule, control code, meetings and peer reviews).
    - **Create, update and execute Verification** **Plan** (white and black test, cyclomatic complexity index calculation, Integration testing, throughput, RAM and FLASH measurement, C99, C11 or other standard evaluation).
  + **SW modules estimated**
    - **Software** **Modules** (RAM, ROM and throughput). Time estimated for each Modules development, it means, they need to reflect the time for every task needed to implement each module like: (UART, I2C or SPI, ADC, PWM, HMI, PID Algorithm implementation, Operative system implementation, etc).
  + **Hw Modules estimated**
    - **Hardware modules** (devices like pc, debugger, board, plant, etc).

# Planning

* It **SHALL** contain the roll definitions of the team members and their responsibilities.
* All the tasks from estimates section **SHALL** be reflected into the Schedule and assigned to the team. Every task **SHALL** contain the definition of done.
* Remember that any document created, updated, White/Black test execution, Integration testing execution, meetings etc, **SHALL** be contained in this section as part of the activities of the plan.

This section **MUST** be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\2) Planning\7. Planning\_20190405.xlsx

# Solving Problem Strategy

* This section SHALL contain an FMEA for the full Project considering the sw functionalities defined in the risk analysis from the estimates section.
* In case an error be detected during the development stage, this section SHALL contain a mitigation plan including the 5 whys methodology for hw, sw and document issues.

This section MUST be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\2) Planning\8. DFMEA\_20190405.xlsx

# Design

This section **SHALL** contain Static and dynamic modeling diagrams like: block diagram flow diagram, call tree diagram, state machine diagram, sequence diagram and others depending on the programming paradigm.

Additionally, this section SHALL contain control diagram where is defined the: inputs, outputs, noise, and its feedback (if apply).

This section MUST be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\3) Design\9. SoftwareDesignDocument\_20190405.docx

Sections 9.1. and 9.2. MUST be contained in this document or MUST be divided into different documents. With the naming defined in every section.

## Standards

In case C89-C90, C11 or other standard be used, it SHALL be specified in this section and additionally add the link to the standard used.

Additionally, the tool used to evaluate the standard SHALL be defined here if apply.

This section MUST be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\3) Design\ 9.1. SoftwareStandards\_20190405.docx

## Naming conventions

The tags SHALL be defined for: local and global variables, local and global functions, macros, enumerations and structures.

It is **SUGGESTED** to use capital letter for global variables and macros.

In case prefix be used, it is SUGGESTED to consider for variable type, module or file, for example:

uint8\_var1

adc\_variable1

etc.

File names SHALL have a convention defined in this section, for instance: first letter SHALL be capital.

For folder in code, it SHALL be defined the names or conventions used.

This section MUST be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\3) Design\9.2. NamingConventions\_20190405.docx

In code comments, It SHALL contain the requirements which is implemented with the code described.

# Testing

## Verification strategy (black box test)

This section SHALL be contained at:

<PROJECT\_PATH>\4) Verification\10.1. BlackboxTest\_baseline.docx

…and its results SHALL be located with the date as suffix, as following is indicated:

<PROJECT\_PATH>\4) Verification\Results\10.1. BlackboxTest\_20190405.docx

Every time a module or feature is implemented, it SHALL contain their tests section and SHALL be contained with the reference to the requirement number in order to have traceability.

## White box strategy

It SHALL define the software which is going to be used, for instance: gtest, junit, sunit, etc.

A document baseline SHALL be created as a reference for all the project implementation. This document SHALL be located at:

<PROJECT\_PATH>\4) Verification\10.2. WhiteboxTest\_baseline.docx

…and its result SHALL be located at:

<PROJECT\_PATH>\4) Verification\Results\10.2. WhiteboxTest\_20190405.docx

Every time a module or feature is implemented, every test case SHALL contain a reference to the requirement number in order to have traceability.

## Cyclomatic Complexity Redundance index

<This section is optional>

This section MUST be contained in this document or in a different document linked to this section, the new document SHALL be contained at:

<PROJECT\_PATH>\4) Verification\ 10.3. CCRI\_20190405.docx

…in case this section is implemented, then its result SHALL be located at:

<PROJECT\_PATH>\4) Verification\Results\10.3. CCRI\_20190405.docx

# Release

Firmware version number SHALL be defined in this section, and the strategy used for that, an example MUST be:

Naming convention for delivered work products like: code and documents shall be defined in this section, the name shall be kept for those documents that SDP describes.

Date/Hw version/Sw version

20190405/001/ 001

The code shall be controlled in GITHUB and path shall be defined here.

## Software Development Folder

The path for software development folder shall be defined in this section and be contained and controlled at GITHUB previous to the final release.

## Integration Tests Strategy

This section SHALL be contained in the planning and reflected in the schedule.

IT **SHALL** be defined a document baseline as a reference for all the project implementation. This document **SHALL** be located at:

<PROJECT\_PATH>\4) Verification\11.1. IntegrationTesting\_baseline.docx

…and its RESULT SHALL be located at:

<PROJECT\_PATH>\4) Verification\Results\11.1. IntegrationTesting\_20190405.docx

Every time a module or feature is implemented, every test case SHALL contain a reference to the requirement number in order to have traceability.

This test MUST contain the plant connected or not.

## Validation Testing / Functional Testing

This section SHALL be contained in the planning and reflected in the schedule.

IT **SHALL** be defined a document baseline as a reference for all the project implementation. This document **SHALL** be located at:

<PROJECT\_PATH>\4) Verification\11.2. ValidationTesting\_baseline.docx

…and its RESULT SHALL be located at:

<PROJECT\_PATH>\4) Verification\Results\11.2. ValidationTesting\_20190405.docx

Every time a module or feature is implemented, every test case SHALL contain a reference to the requirement number in order to have traceability.

This test SHALL contain the plant connected.

## Throughput and Flash and RAM measurement

This section SHALL be contained in the planning and reflected in the schedule.

It SHALL define the RAM, Flash and Throughtput measurements strategy at:

<PROJECT\_PATH>\4) Verification\ 11.3. ThroughputRAMFlash\_procedure

# Results

All pictures, videos or miscellaneous SHALL be posted at:

<PROJECT\_PATH>\5) Results

# Lessons Learned

All comments, feedback or others SHALL be documented in this section.