**Audio Player Middleware Component Development**

| Author | Prachi Verma |
| --- | --- |
| Version | V1.0 |
| Date | 12-01-2023 |

Revision History

| Date | Change | Author | Version |
| --- | --- | --- | --- |
| 11-01-2023 | Documentation not added | Prachi Verma | V1.0 |
| 12-01-2023 | Static Analysis added | Prachi verma | V1.0 |
|  |  |  |  |

Contents

[**1.**](#_heading=h.gjdgxs) **Introduction to Software Development Process** 3

[**2.**](#_heading=h.30j0zll) **Software Requirement** 4

[**3.**](#_heading=h.1fob9te) **Software Design** 5

[2.1 High Level Design 5](#_heading=h.3znysh7)

[2.1.1 Objective 5](#_heading=h.2et92p0)

[2.1.2 Block diagram and components 5](#_heading=h.tyjcwt)

[2.2 Low Level Design 6](#_heading=h.3dy6vkm)

[2.2.1 Defines 6](#_heading=h.1t3h5sf)

[2.2.2 Data Structure 7](#_heading=h.4d34og8)

[2.2.3 APIs 7](#_heading=h.2s8eyo1)

[2.2.4 State Machine 9](#_heading=h.17dp8vu)

[**4.**](#_heading=h.3rdcrjn) **Software Development/Implementation** 10

[**5.**](#_heading=h.26in1rg) **Software Verification and Validation** 11

[5.1 Verification 11](#_heading=h.lnxbz9)

[5.1.1 Static Analysis 11](#_heading=h.35nkun2)

[5.1.1 Unit Testing 11](#_heading=h.1ksv4uv)

[5.1.1.1 Test cases 11](#_heading=h.44sinio)

[5.1.1.2 Test Results 11](#_heading=h.2jxsxqh)

[5.1.1 Review 12](#_heading=h.z337ya)

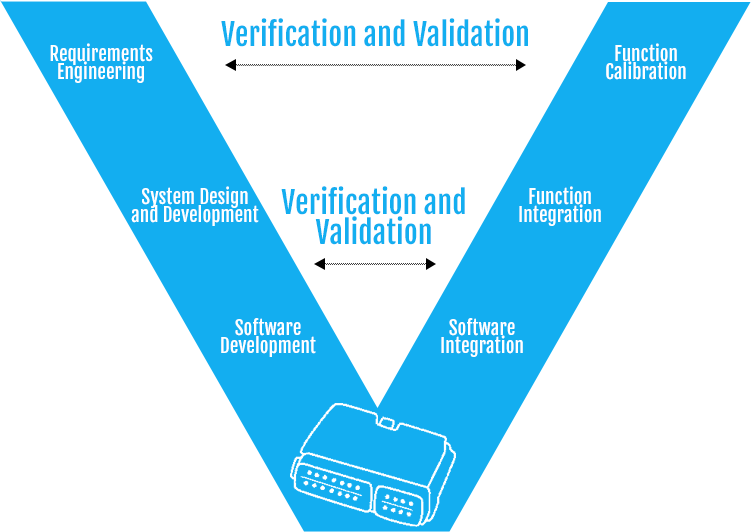
[5.2 Validation 12](#_heading=h.3j2qqm3)

[5.2.1 Functional Testing 12](#_heading=h.1y810tw)

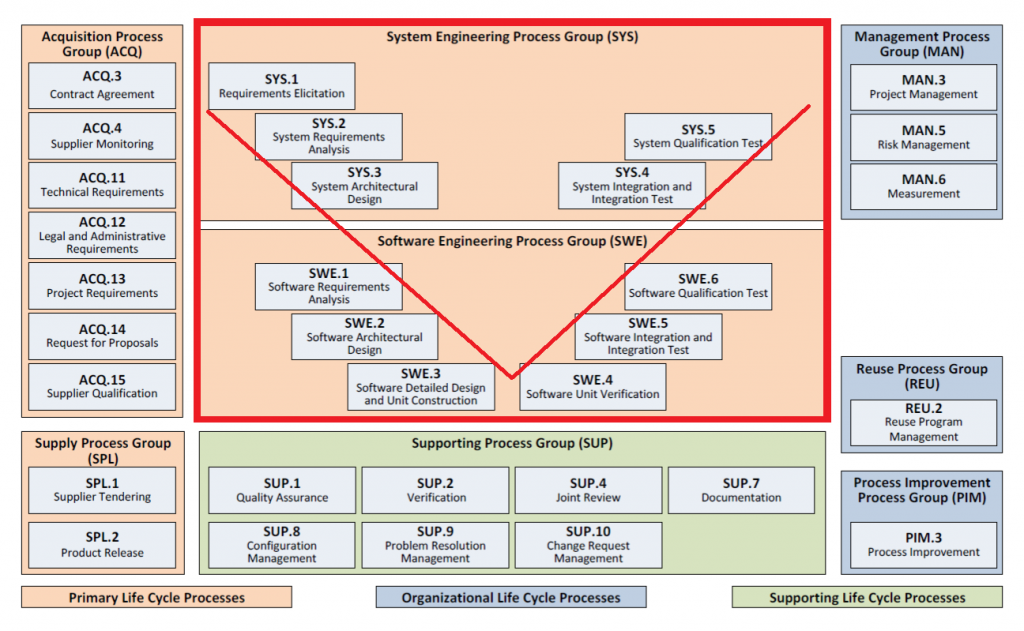
[5.2.2 Integration Testing 12](#_heading=h.4i7ojhp)

[5.2.3 System Testing 12](#_heading=h.2xcytpi)

# **Introduction to Software Development Process**



Software Development Process ( V-Model)



Software Development Process – Automotive SPICE

# **Software Requirement**

* APLAYER\_R\_001: To develop a music player middleware API to HMI
* APLAYER\_R\_002: It shall provide an API to select an audio file source
* APLAYER\_R\_003: It shall provide an API to play an audio file source
* APLAYER\_R\_003: It shall have an API to stop an audio file source

# **Software Design**

This section describes the High-Level design of mplayer module.

## 2.1 High Level Design

## 2.1.1 Objective

<Write down the objective>

## 2.1.2 Block diagram and components



<List down the components and give a short description>



<List down the components and give a short description>

## 2.2 Low Level Design

This section provides details of Low level design.

## Defines

#define ***PLAYER\_SOURCE\_LEN 128***

Defines key event

**typedef enum {**

AKEY\_STOP,

AKEY\_START,

AKEY\_INVALID

**}PlayerKey\_e;**

Defines player states

**typedef enum {**

ASTATE\_INIT,

ASTATE\_IDLE,

ASTATE\_PLAYING

**} PlayerState\_e;**

## Data Structure

***struct PlayerSource\_st {***

*char strSource[PLAYER\_SOURCE\_LEN];*

*int nCurrentOffset;*

***}***

***struct Player\_st {***

*PlayerState\_e eCurrentState;*

*struct PlayerSource\_st stPlayerSource;*

**}**

## APIs

Syntax : **Player\_st \*PlayerInit(void);**

Parameters (in) : none

Parameters (out) : none

Return value :Player\_st\*

Description : this function will allocate memory dynamically

initialize eCurrentState .

Syntax :**struct Player\_st \*PlayerSelectSource( struct Player\_st \*pstPlayer, char \*strSource)**

Parameters (in) : Player pointer, source file name.

Parameters (out) : it will print which source source file is selected

Return value : it will return pstPlayer pointer to structure

Description : the function will take player structure and file name and assigne it to strSource n variable of stPlayerSource struct .

Syntax : **void PlayerSMHandle(struct Player\_st \*pstPlayer,**

**PlayerKey\_e eKeyPressed)**

Parameters (in) : player\_st structure and eKeyPressed

Parameters (out) : based on the key user pressed it will print wether song

is playing or it is paused.

Return value : void

Description : This function will take player\_st structure and key

pressed by user and it will asssign this state

eCurrentState depending upon what state user gave.

Syntax :  **void PlayerDeInit(struct Player\_st \* pstPlayer)**

Parameters (in) : pstPlayer structure

Parameters (out) : none

Return value : void

Description : this function will free the player structure memory

allocated when we called init function.

## State Machine



* Pseudo-code
* Files Organizations

# **Software Development/Implementation**

See the source code files.

# **Software Verification and Validation**

## Verification

## 5.1.1 Static Analysis

| File Name/Line Number | MISRA C Error | Status | Fix | Comment |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 5.1.2 Unit Testing

## 5.1.2.1 Test cases

| TestID | Test Case | Pre-condition | Steps | Expected Results |
| --- | --- | --- | --- | --- |
| 1. | Select a .mp3 file | Check if file is .mp3 or not | 1.Take input as a string.  2.String must contain .mp3  3.Accept the file  4.Else reject | if file given have .mp3 then it will work and ask for other options. |
| 2. | Use Player Or not | To Use the player you have to choose 1. | 1.Take input as 1 to use player. | Player will start |
| 3. | Stop Player | choose any value other than 1 | press any value | player will stop  and prints error. |

## 5.1.2.2 Test Results

| TestID | Test Case | Pre-condition | Steps | Expected Results | Actual Results | PASS/FAIL |
| --- | --- | --- | --- | --- | --- | --- |
| 1. | Select a .mp3 file | Check if file is .mp3 or not | 1.Take input as a string.  2.String must contain .mp3  3.Accept the file  4.Else reject | if file given have .mp3 then it will work and ask for other opti | .mp3 | PASS |
| 2. | Use Player Or not | To Use the player you have to choose 1. | 1.Take input as 1 to use player. | 1.Take input as 1 to use player. | Player will start | PASS |
| 3. | Stop Player | choose any value other than 1 | press any value | player will stop  and prints error. | player will stop | PASS |

## 5.1.3 Review

Review records

| File Name/Line No | Review comment | Status | Comments |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

## Validation

## Functional Testing

## Integration Testing

## System Testing