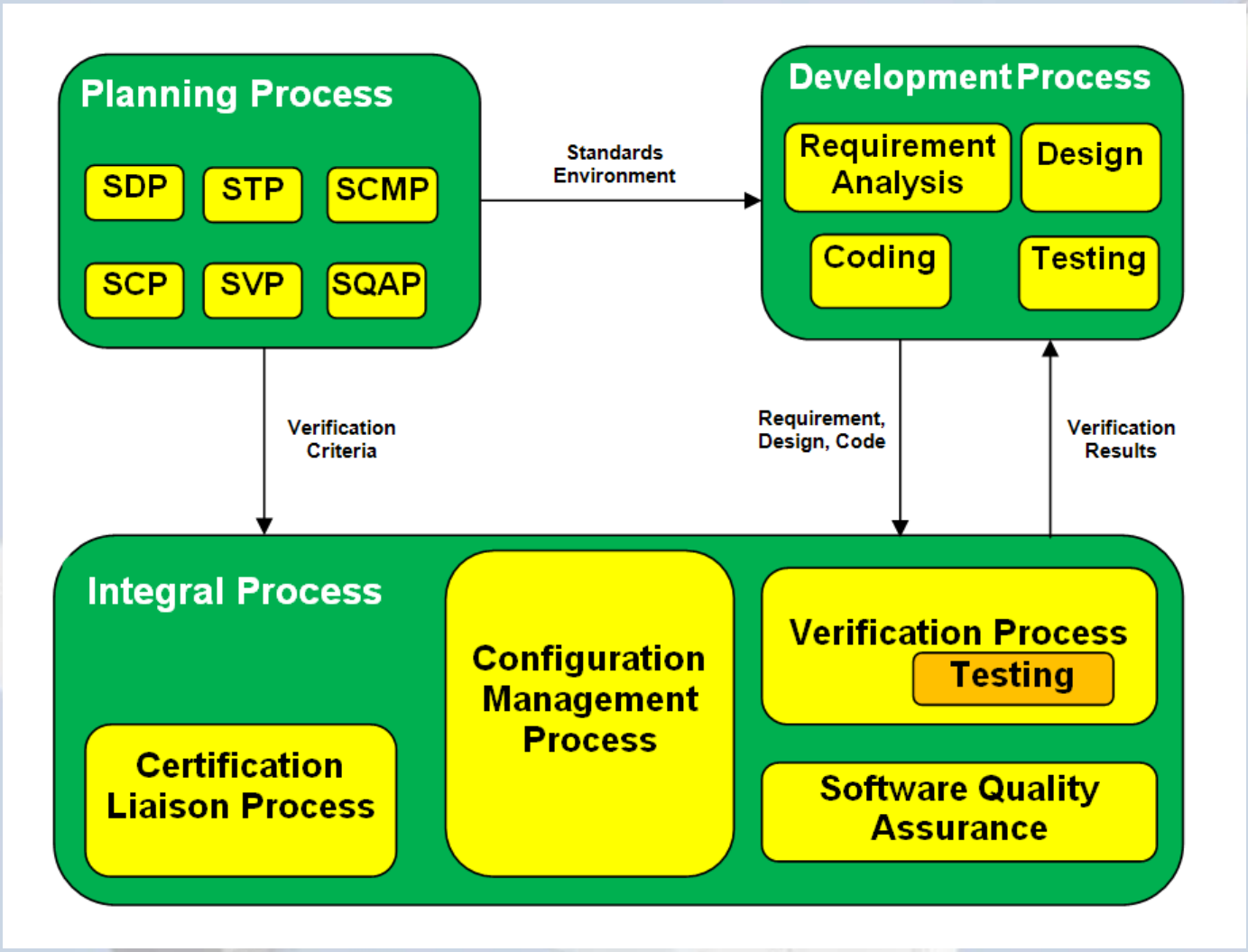
CASE STUDY
SOFTWARE DEVELOPMENT USING FORMAL METHODS @ MCSRDC

Software Development Life Cycle

- SDLC at MCSRDC, HAL is based on
  - DDPMAS (Procedure for Design, Development and Production of Military Aircraft and Airborne Stores) "Software Development and Certification" released by CEMILAC.

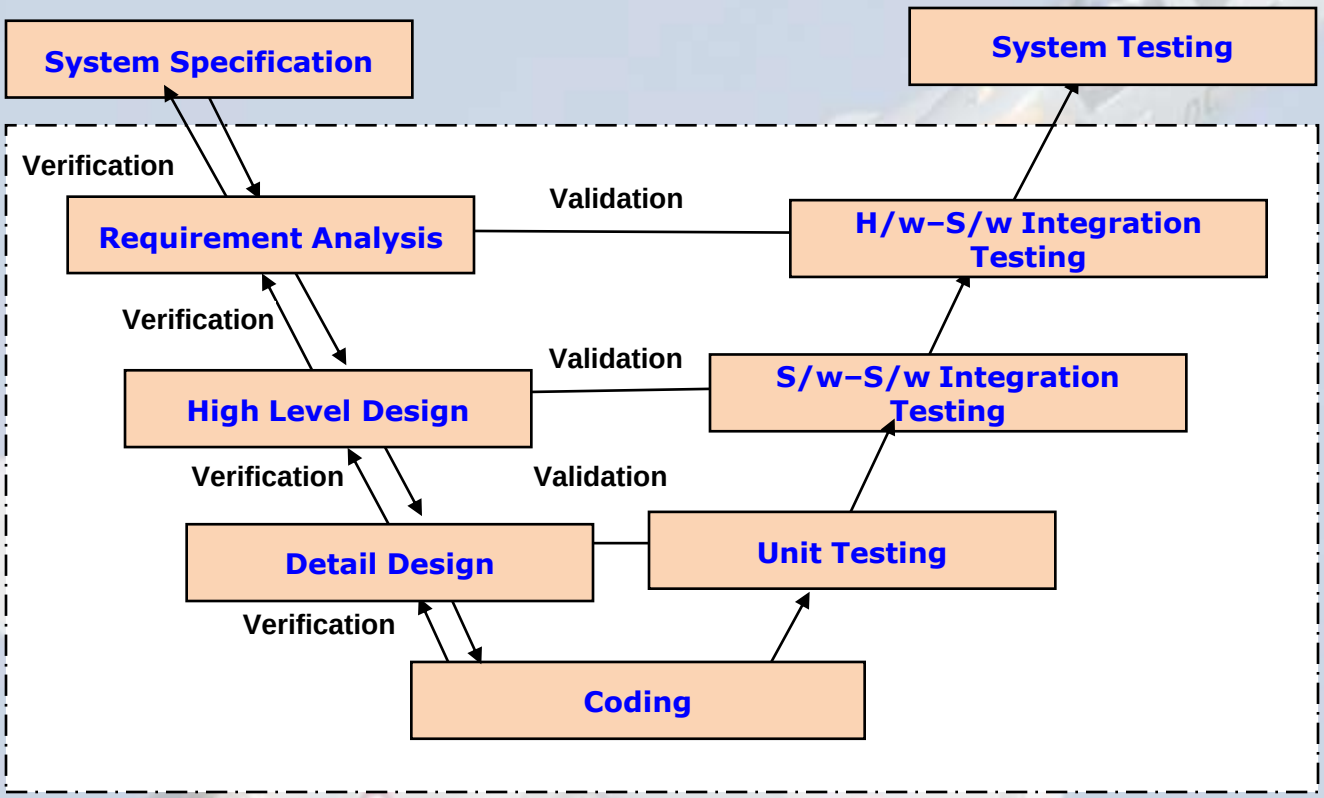
- Types of Software which are used in the aircraft and its support systems
  - Airborne Embedded Software
  - Ground / Test Equipment Software
  - Test Rig Software

Software Development Life Cycle at MCSRDC



Software Development Process – V Model

- Suitable for the systems for which requirements are of high level of assurance and does not change during development.



Software Planning Process

Input

- System Specification
- Project Plan

Task

- Organization of S/W Development Team
- Analyzing Project Requirements & Schedules
- Analyzing S/W scope and complexity levels
- Decision on S/W development Language, Tools and Development Environment
- Development Activities with Time Line
- Risk Management
- H/W and S/W Infrastructure
- V&V, SCM , SQA methods and route to certification

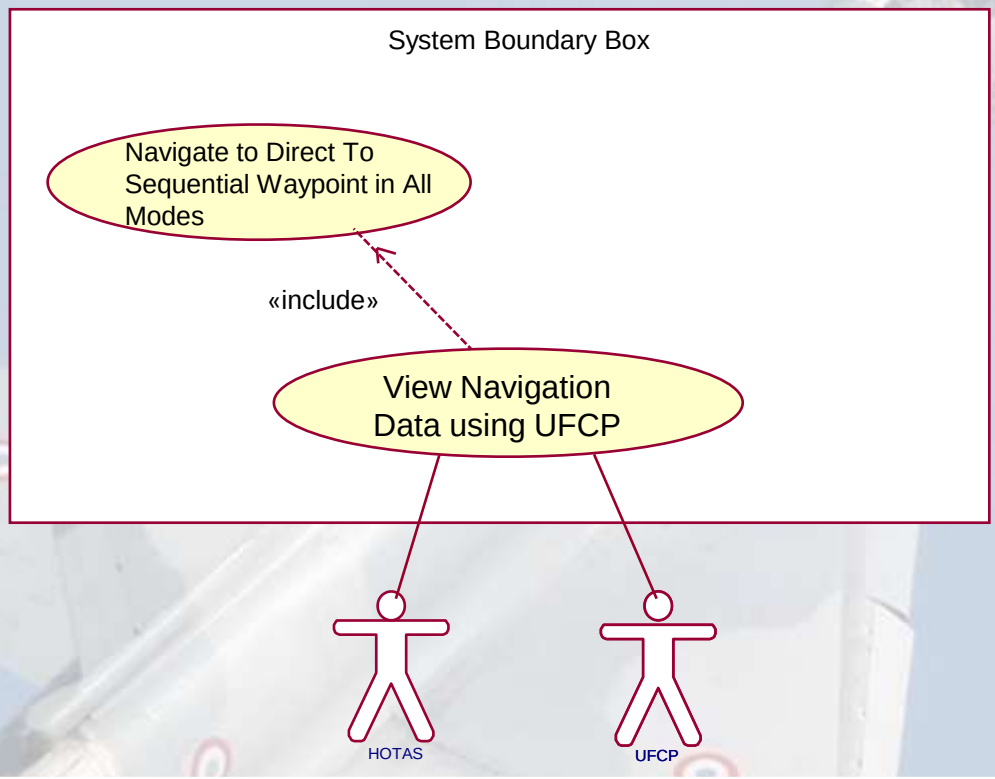
Output

- SDP
- SCMP
- SVP
- STP
- IV&VP
- SCP

Requirement Analysis Process

Requirement Analysis Process	
Entry Criteria	• System Specification document is baselined for Requirement Analysis.
Inputs	• System Specification, System level requirements • Software ICD • Software Development Plan • Derived documents (FDA, BIT etc.)
Tasks	• Identification of use cases • Identification of Actors and generation of Actor catalogue • Prepare Context Diagram • Prepare Hardware Architecture Diagram • Prepare Constraint Diagram and Description • Prepare System State Diagram • Analysis of Use Case interaction with actors and other use cases through Use case Diagrams. • Preparation of Traceability of SRS to System Spec and reverse to SRS from System Specification • Submission of SRS for configuration management as developmental baseline
Verification	• Review of Traceability Matrix • Internal Verification and Validation of SRS • Independent Verification and Validation of SRS • Process Verification
Output	• Software Requirement Specification • Traceability Report • SRS Review sheet reports
Exit Criteria	• Release of SRS after approval.

View Navigation Data using UFCP



USE CASE: View Navigation Data using UFCP

Description:
The functionality of this use-case is to display navigation parameters (WP No., FPL no., distance, homing, Direct Track (DTK), Track Made Good (TMG), Planned Track (PTK), Ground Speed (GS), Running Time (RT), Computed Ground Speed (CGS), Time-To-Go (TTG), ELY, LTE, ahead or behind information i.e. Along Track Distance and Displacement from the track i.e. Cross Track Deviation etc.) on UFCP using different Nav pages. If an attack mode is engaged, the mode is also displayed.
Refer:
System Specification for Jaguar DARIN Upgrade-DARIN-III section 3.1.4.3.5

Constraints

NIL

Availability

PRIMARY, REV1 and REV2 modes

Entry condition for Use Case

1.Pressing enter in normal alignment status page 2 or in reversionary NAV operation page
2.When alignment status changes to ‘ready’ after alignment.
3.Selection of NAV option from Top Level Menu Page.
4.By conditionally exiting from page selected through joystick from this page.
5.If any active Push button is depressed from this page, then on exit from that selected active button page, this page will be reselected.
6.when time to go to ‘WP’ is 70s (Altitude <6000’) or 130s (Altitude >6000’) in SEQ NAV mode and weight on wheel is ‘off’ (air).
7.Pressing S8 switch Down on Sticktop in WD/WS page.
8.Pressing S9 switch Centre on Sticktop in Top Level Menu page.

Use Case Description

Functionality Requirements:

Spec Ref.	Req. No.	Description
[DARINIII_SPEC] 3.1.4.3.5, 3.1.4.3.5.1, 4.9.1, 5.1.3.1, 5.2.3.1	R 1	Nav Page, DTK/TMG Page, PTK/TMG Page, Time Page, Close Nav Page and Wd/Ws Page shall be selected in cyclic manner using Scroll switch or down key of five-position switch (S8) on HOTAS.
[DARINIII_SPEC] 3.1.4.3.5.1	R 2	PTK/TMG page shall not be available in manual mode and while scrolled in sequential order the page next to PTK/TMG shall be selected.
[DARINIII_SPEC] 3.1.4.5	R 3	The engaged attack mode (A/A, CCIP, CCRP, OPP, and AGG) shall be displayed on Navigation pages on last three column of second row (2x22).
[DARINIII_SPEC] 3.1.4.3.5.5, 3.1.4.5	R 4	If close navigation condition (time to go less than 70s/130s depending upon height being less than or greater than 6000 ft.) is met in SEQ Nav mode or during CCIP/CCRP/OPP attack and WOW is off (in air), the Close Nav Page will be displayed on UFCP.
[DARINIII_SPEC] 3.1.4.5.1, 3.1.4.5.2, 3.1.4.5.3, 3.1.4.5.4, 3.1.4.5.5, 3.1.4.5.6	R 5	Homing shall be displayed at 1x1 and Distance to next Way Point (WP) shall be displayed at 1x11 in default Nav, DTK/TMG Page, PTK/TMG Page and Time Page, and in Close Nav Page it is at 1x10.

Use Case Description

[DARINIII_SPEC] 3.1.4.5.1, 3.1.4.5.2, 3.1.4.5.3, 3.1.4.5.4,3.1.4.5.5, 3.1.4.5.6	R 6	Active Flight Plan Number shall be displayed at 5x1 and next Way Point number shall be displayed at 5x11 in Nav page and all its sub pages except close nav page where it is 5x10.
[DARINIII_SPEC] 3.1.4.3.5.1,3.1.4.3.5.2, 3.1.4.3.5.3, 4.9.1, 5.1.3.1, 5.2.3.1	R7	DTK shall be displayed at 3x11 in DTK/TMG Page. PTK shall be displayed at 3x11 in PTK/TMG Page. TMG shall be displayed at 3x1 in Nav page, DTK/TMG page and PTK/TMG page.
[DARINIII_SPEC] 3.1.4.3.5.2, 3.1.4.3.5.3, 3.1.4.3.5.4, 3.1.4.3.5.5, 4.9.1, 5.1.3.1, 5.2.3.1	R8	Displacement from the track shall be displayed at 2x14 in DTK/TMG Page, PTK/TMG Page and Time Page. If the aircraft is to the left of the track 'L' shall be displayed at 2x11. If the aircraft is to the right of the track 'R' shall be displayed at 2x11. Displacement from the track shall be displayed at 2x13 in close nav page and 'L' or 'R' shall be displayed at 2x10.
[DARINIII_SPEC] 3.1.4.3.5.2, 3.1.4.3.5.3, 3.1.4.3.5.4	R9	A/C ahead or behind information (A/B) shall be displayed at 2X1 in DTK/TMG Page, PTK/TMG Page, Time Page, and Close Nav Pages. This indicates the computed along track distance of the aircraft on the current flight leg.

Use Case Description

[DARINIII_SPEC] 3.1.4.3.5.4,3.1.4.3.5.5, 4.9.1, 5.1.3.1, 5.2.3.1	R10	Running Time (RT) shall be displayed at 3x11 in Time page and it is set by pilot in clock setting page. Ground Speed (GS) shall be displayed at 3x11 in Nav page and 3x2 in Close Nav and Time Page.
[DARINIII_SPEC] 3.1.4.3.5.5	R11	In Close Nav Page, TTG (Time To Go) shall be displayed at 3x10 as HH:MM:SS and during attack mode TTR (Time to Release) will be shown in place of TTG (Time to Go).
Derived Requirement	R12	Navigation Parameters that are displayed on the pages are received from Navigation package. Certain parameters that are specific to a page are computed by that particular page, e.g. Commanded Ground Speed (CGS), Expected Time of Arrival (ETA), Early (ELY) and Late (LTE). The formulae for computation of these parameters are given as CGS = (distance to go)/ (PTA - RT). ETA = (RT - time to go) ELY/LTE = PTA – ETA DTK, PTK and Cross Track Distance shall be displayed as blank for first WP of FPL.
[DARINIII_SPEC] 3.1.4.3.5.1, 3.1.4.3.5.2, 3.1.4.3.5.3	R13	HUD and EXIT keys will always be active in all Nav pages to facilitate viewing of REV DSL aiming mark and TLM (Top Level Menu) page.

Special Requirements:
Not Applicable

Activity Diagram:
Not Applicable

Availability of Entered/Selected Parameter
Nil

Final Effect of Entry/Selection in other functionalities
This Use-Case is for viewing data on UFCP.

Exit Condition for Use Case
Pressing EXIT will lead to Top Level Menu Page of UFCP.

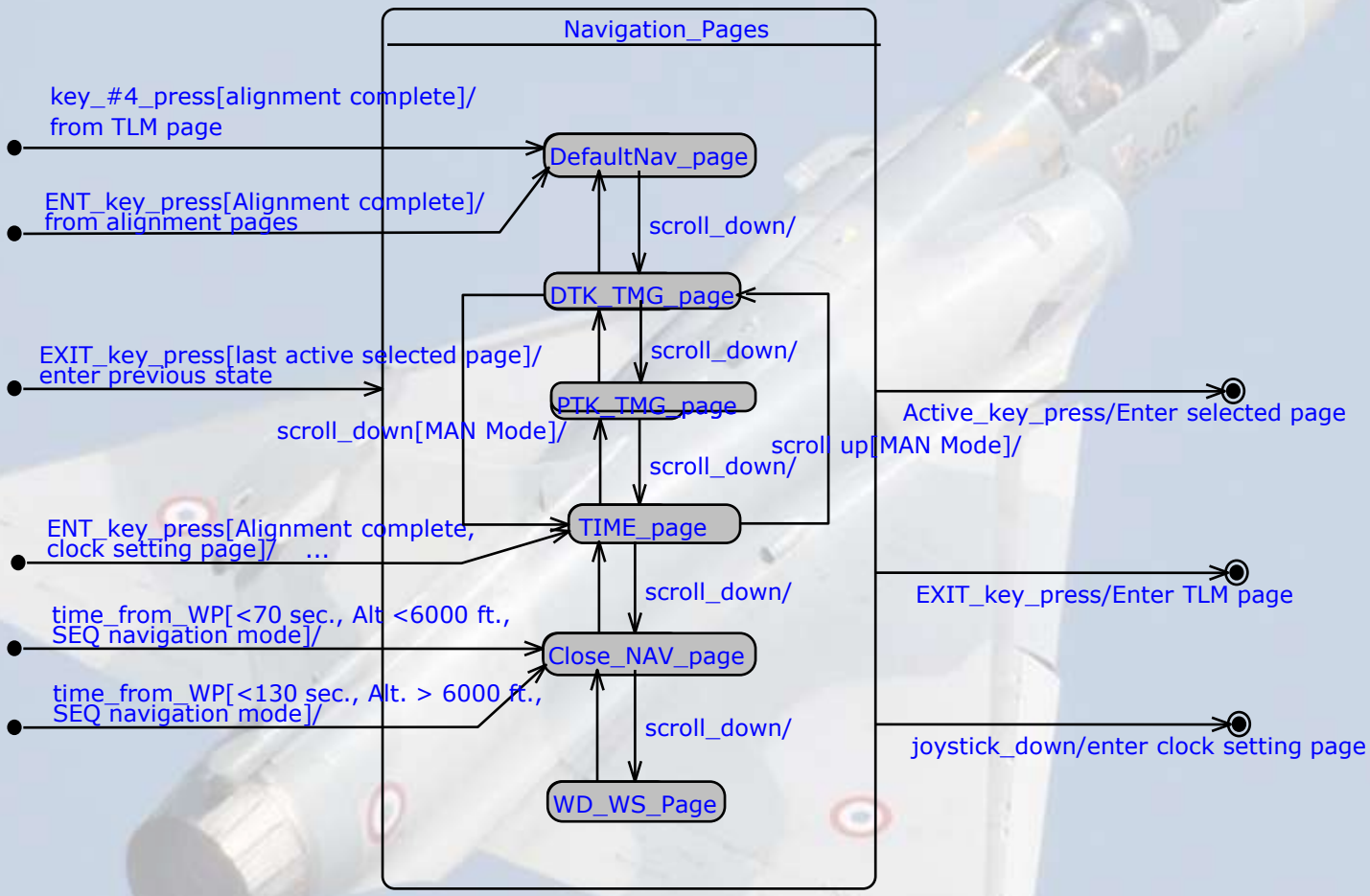
Use Case interaction with actors

UFCP - It is used to display navigation parameters like Homing/ Distance/ FPL No/ WP No/ PTA/ DTK/ Time to go/ Early / Late/ WD/WS etc and it also shows attack mode engaged. These informations are displayed on UFCP using different nav pages.
HOTAS - It can be used to navigate between different Nav pages.

Software Design Process

Software Design Process	
Entry Criteria	• Software Requirements are identified and baselined for development
Inputs	• Software Requirement Specification (SRS)
Tasks	**For OOAD:** • Identify Classes and member functions • Prepare Sequence Diagrams • Prepare Class Diagrams • Provide definition for the Class operations and attributes • Documenting operations and attributes **For SOAD:** • Identify Modules and associated functions • Prepare File diagrams, Message diagrams and Call Graphs • Provide definitions of functions and pseudo code and global variables **Common to Both OOAD and SOAD:** • Prepare State transition diagram • Algorithm development for any special algorithm intensive requirement • Prepare Test Procedure(s) • Preparation of Traceability of SRS to SDD • Submission of SDD for configuration management as developmental baseline
Verification	• Review of Traceability Matrix • Internal Verification and Validation of SDD. • Independent Verification and Validation of SDD. • Process Verification
Output	• Software Design Document (SDD) • Traceability Report • High Level Design and Detailed Design Review sheets
Exit Criteria	• SDD is approved and Released

STATE TRANSITION DIAGRAM



SEQUENCE DIAGRAM

Call the service with active page
If active_page_m is in-flight alignment page
Set, ENTER, EXIT and the number function keys as the only active keys for the page
End if
Call SetPageContext
Call Build Page
Prepare data to be displayed on the page
Call Update page data with latest data to be displayed

Call Get Active FPL number

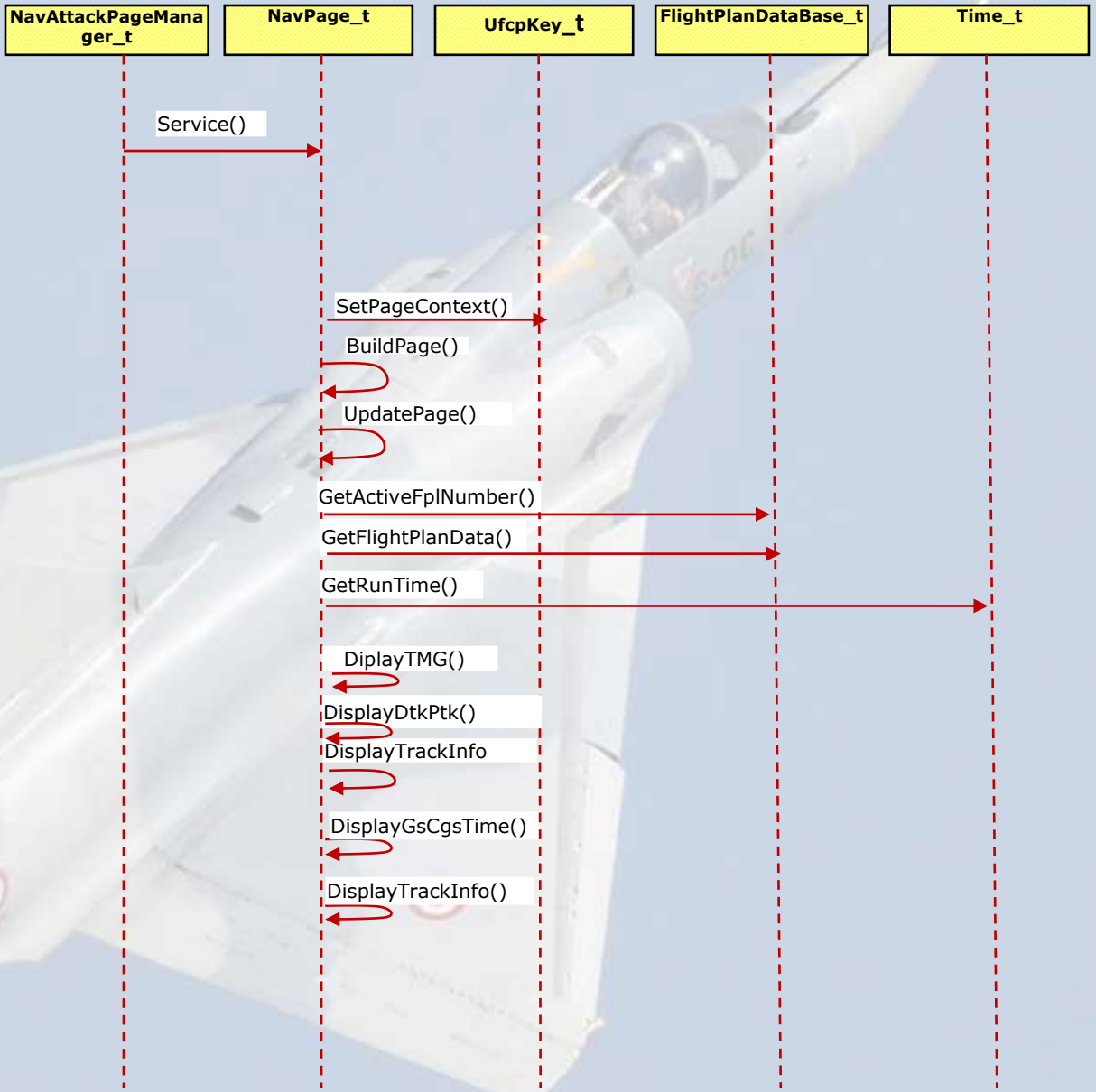
Call Get flight plan data of the current FPL

Call Get run time
If active page is DTK TMG PAGE or PTK TMG PAGE then
Call Display TMG

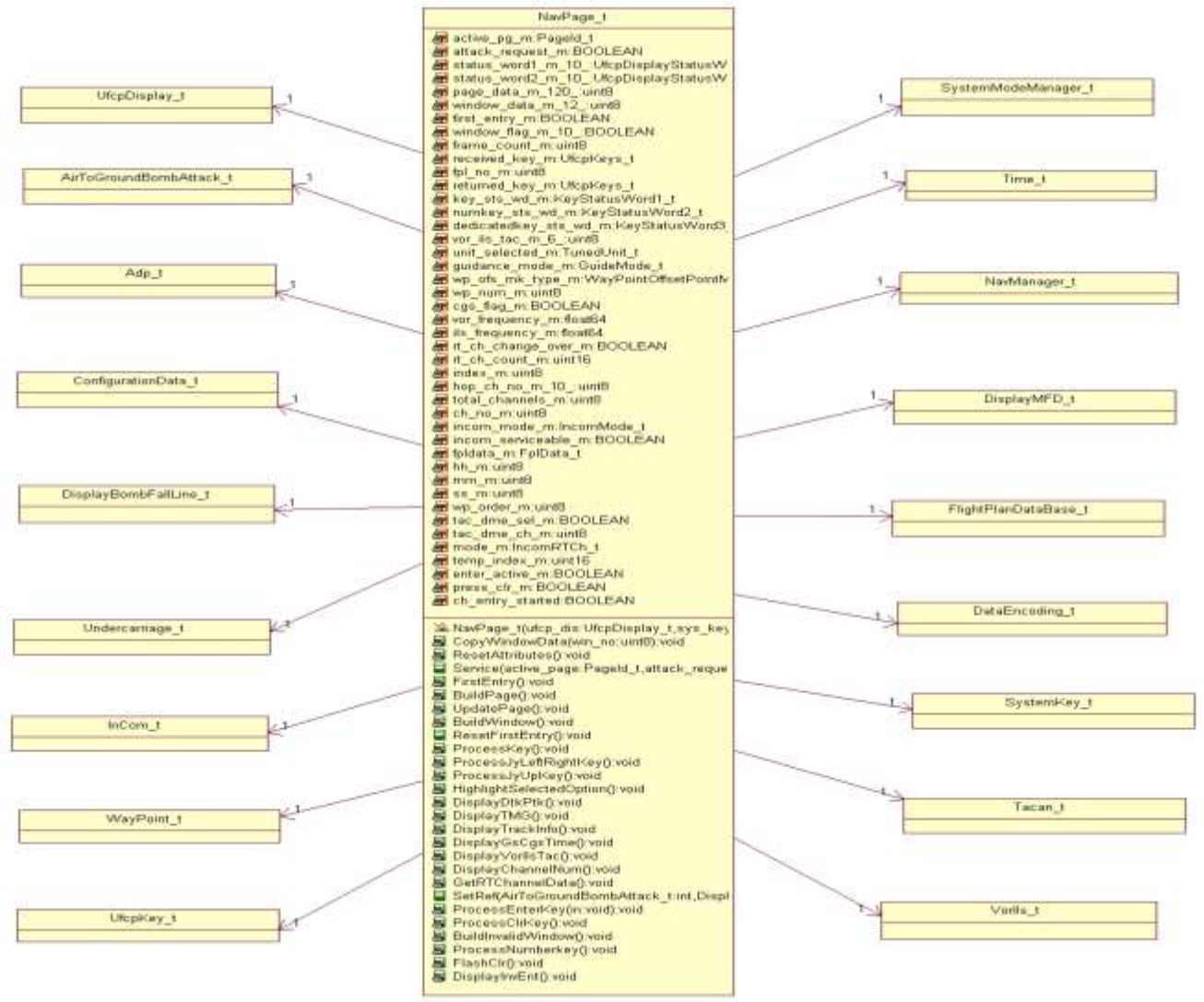
Call Display Dtk/Ptk

Call Display Track Info
Else if active page is TIME PAGE or CLOSE NAV PAGE then
Call Display GsCgsTime

Call DisplayTrackInfo
Else
Display default NAV page (NO MESSAGE)



CLASS DIAGRAM

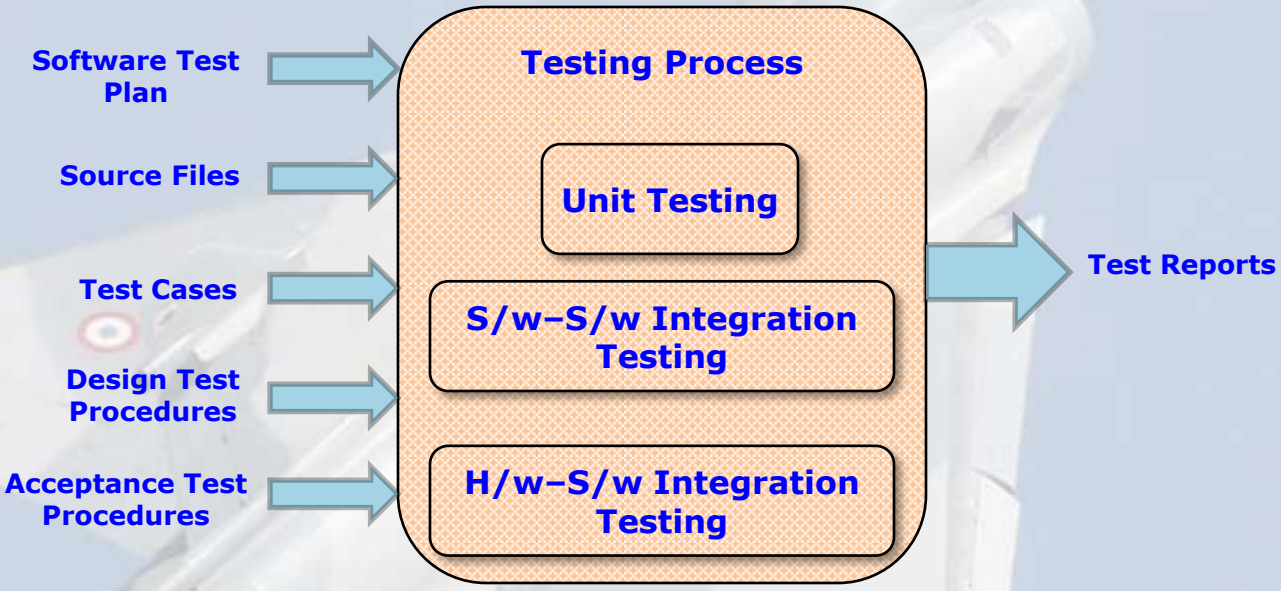


Implementaion Process

Software Implementation Process	
Entry Criteria	• Software Designs are baselined for development
Inputs	• SDD • Code of Practice for Implementation Language as mentioned in SDP • Derived Documents (e.g. FDA, BIT etc.)
Tasks	• Create Header File(s) ➢ Define Class ➢ Add Public and private members of class ➢ Define Data Type of attributes ➢ Define Prototype for operations ➢ Add Description for attributes and operations • Create source File(s) ➢ Include respective Header File ➢ Add Operation Description ➢ Provide operation description as mentioned in SDD • Compile the Build • Preparation of Traceability of Code to SRS and SDD to code • Verification of Code at author level
Verification	• Internal Verification and Validation of source files and header files • Independent Verification and Validation of code • Review of Traceability Matrix • Process Verification • Submission of Source files for configuration management
Output	• Source Code and executable files • Code Review sheets
Exit Criteria	• Source Code and Executable are Approved and Released.

Software Testing Process

Purpose of testing is to demonstrate that the software satisfies the requirements and to demonstrate with a high degree of confidence that it does not contain any features which could result in erroneous operations.



Tools used during testing:

- LDRA
- RT Simulators, Sensor simulators
- System and Software Rigs

Activities:

- **Unit Testing** : Ensures that Class /function meets intended Design.(Static and dynamic tests Using LDRA)
- **S/w-S/w Integration Testing**: Ensures correctness of interfaces between different Software modules.
- **H/w-S/w Integration Testing** : Ensures correctness of Software with Software requirements on Target Hardware.

Software Testing Process	
Entry Criteria	• Source Files are ready for testing activity
Inputs	• Source Files and Executables • Software Test Plan • Test Cases, DTP and Acceptance Test Procedure(s)
Tasks	• Static and dynamic Testing is performed • Testing software functionality as per Test Cases, DTP and ATP • Submission of the test procedures and test reports for configuration management as developmental baseline
Verification	• Software Requirements are verified and validated • Test cases are verified & validated at unit level for Strutural coverage(statement, branch and mcdc)
Output	• Test Reports, coverage analysis reports • Compiled, Linked Object Code after Testing
Exit Criteria	• Test Reports for all modules are generated and approved.

During Static Analysis the following points are considered

- No unreachable code exists
- No Up-Down or Up-Up Knots present
- McCabe Cyclomatic complexity less than or equal to 10
- Nesting Figure less than or equal to 7
- McCabe Essential Complexity less than or equal to 1
- Essential Knots figure equal to 0
- All declared variables must be used
- No UR(Undefined referenced) anomalies
- No DD(Doubly Defined) anomalies
- No DU (Defined Unused) anomalies

Some typical static testing violation shown by LDRA

- 110 S:-Use of // style of comment instead of /*….*/.
- 56 S:-Equality comparison of floating point numbers.
- 50 S:-Use of shift operator in signed types.
- 59 S:-Else statement missing for if.
- 60 S:-Empty switch statement.
- 61 S:-Switch statement contains default only.
- 62 S:-Switch statement not terminated with break.
- 48 S:-No default statement for switch
- 96 S:-Use of mixed mode arithmetic.
- 434 S:-Signed/unsigned conversion without typecast.

Dynamic Analysis Procedure

- In dynamic analysis the source code to be analyzed is instrumented, compiled, linked and executed
- When the source code has been instrumented, compiled, linked and executed, it performs its normal computations but creates an output stream which contains the execution history
- The Dynamic Coverage Analysis option processes this execution history, mapping its information on to the control flow information on the source code acquired from the Static Analysis phase
- On completing the test Regression Reports(showing the PASS/FAIL reports of the test cases),Dynamic Coverage Analysis Report and tcf files are saved for future reference

Dynamic Analysis Coverage Measures

Dynamic Analysis reports on the effectiveness of test data by means of coverage metrics :-

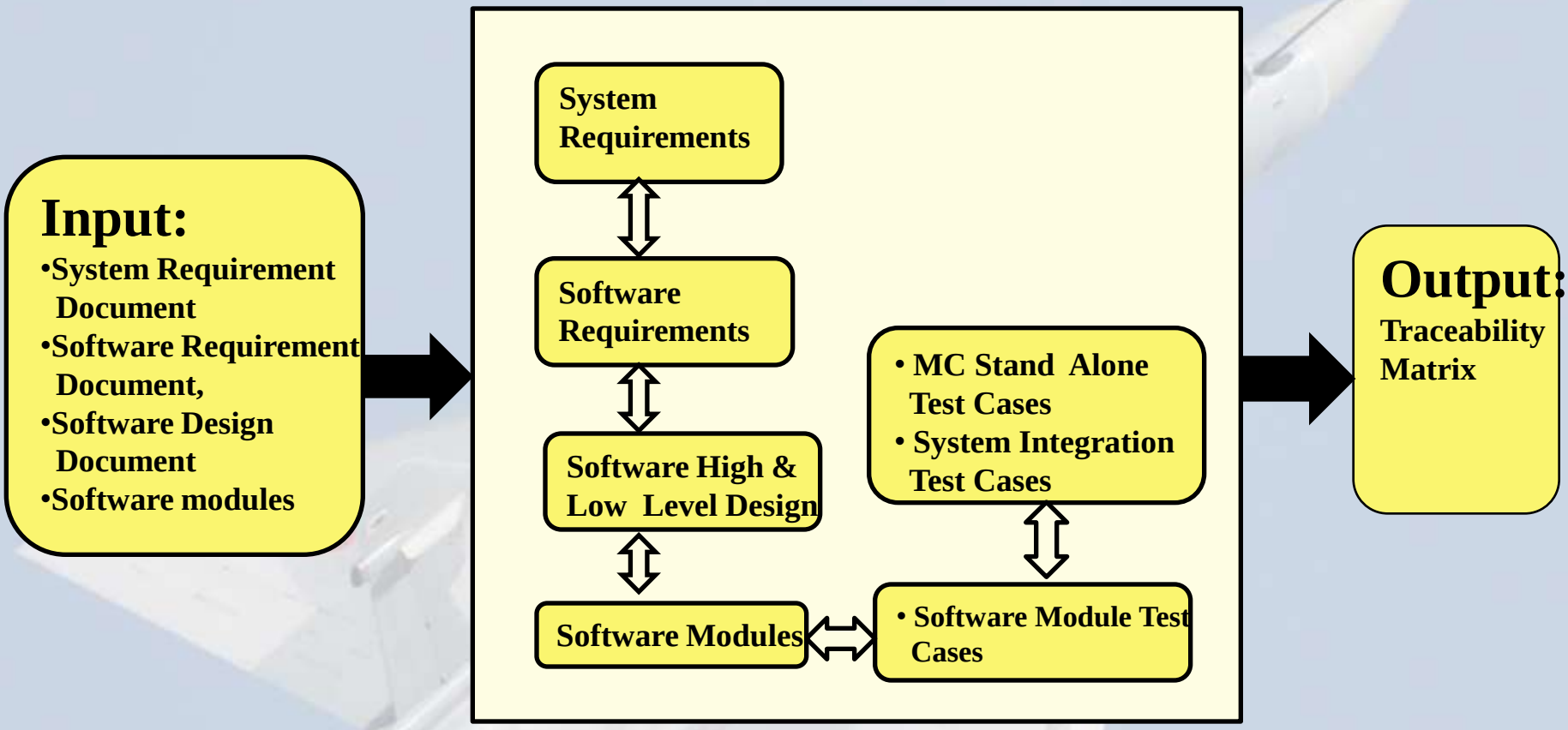
Statement Coverage-TER1

Branch/Decision Coverage-TER2

Modified Condition Decision Coverage-MC/DC

[TER: Test Effectiveness Ratio]

Traceability for entire Life Cycle



Traceability for entire Life Cycle

SL No	System Specification	FDA	Software Requirement Specification		High Level Design		Detail Design		Code		Module Test	SIR Test
			Use Case No/Use Case Name	Req. No	Sequence Diagram	Class Diagram	Class No/Class Name	Function Name	Identifier (File Name, Version, Revision)	Function Name	Module Test Procedure	ATP Section
1	Derived Req	NA	UC 1.25: Display Fuel Calibration Page and Fuel Level setting page on CDU	R1	Fuel Calib and Setting Sequence	Fuel Calib and Setting Class	1.73.2/FuelCalibPage_t	Service()	DMC_MP:FUELCALIBPAGE_CPP. SOURCE;1 V1.001	Service()	TC_FuelCalibandSet.doc	1.25.1
2	Derived Req	NA		R2			1.47.5 / CduKey_t	GetCduKey()	DMC_MP:CDUKEY_CPP.SOURCE;1 V1.001	GetCduKey()		1.25.2
3	A 1.4.2	NA		R3			1.47.3 / DualDiuManager_t	GetOnGroundStatus()	DMC_MP:DUALDIUMANAGER_CPP.SOURCE;1 V1.001	GetOnGroundStatus()		1.25.3
4	A 1.4.2	NA		R4			1.73.2/FuelCalibPage_t	DisplayDefaultPage()	DMC_MP:FUELCALIBPAGE_CPP. SOURCE;1 V1.001	DisplayDefaultPage()		1.25.4
5	A 1.4.2	NA		R5			1.47.6 / CduDisplay_t	DisplaySpecialCharacter() SetTextColor()	DMC_MP:CDUDISPLAY_CPP. SOURCE;1 V1.001	DisplaySpecialCharacter() SetTextColor()		1.25.5

Levels Involved

- Peer Level
- Team Level
- Independent V&V

Types of Analysis

Traceability Analysis:
Traceability matrix is generated to identify any missing High level/low level requirement or any functionality that has not been addressed in design or code.

Software Requirement based coverage analysis
Ensuring that all the stated requirements for the software are tested

Structural Coverage Analysis
The structural analysis identifies the code structures. Dynamic Test results shall be analyzed for code coverage.

Data flow Analysis
Ensuring that all inputs are used, all outputs are properly generated, no variable uninitialized, no local variables unused and all outputs properly generated

Control flow analysis
Control flow within source files can be verified by checking call sequences are as expected if the specific calls have been made during testing

Software Configuration Management Process

- SCM Activities
  - Identification of Software Configuration Items (SCI's).
  - Baseline & traceability establishment.
  - Problem Management.
  - Change Control.
  - Archive, retrieval & release.
  - Environmental Control.
- SCM Reports
  - Problem Reports / Change Requests.
  - Software Configuration Index
  - Software Life Cycle Environment Configuration Index

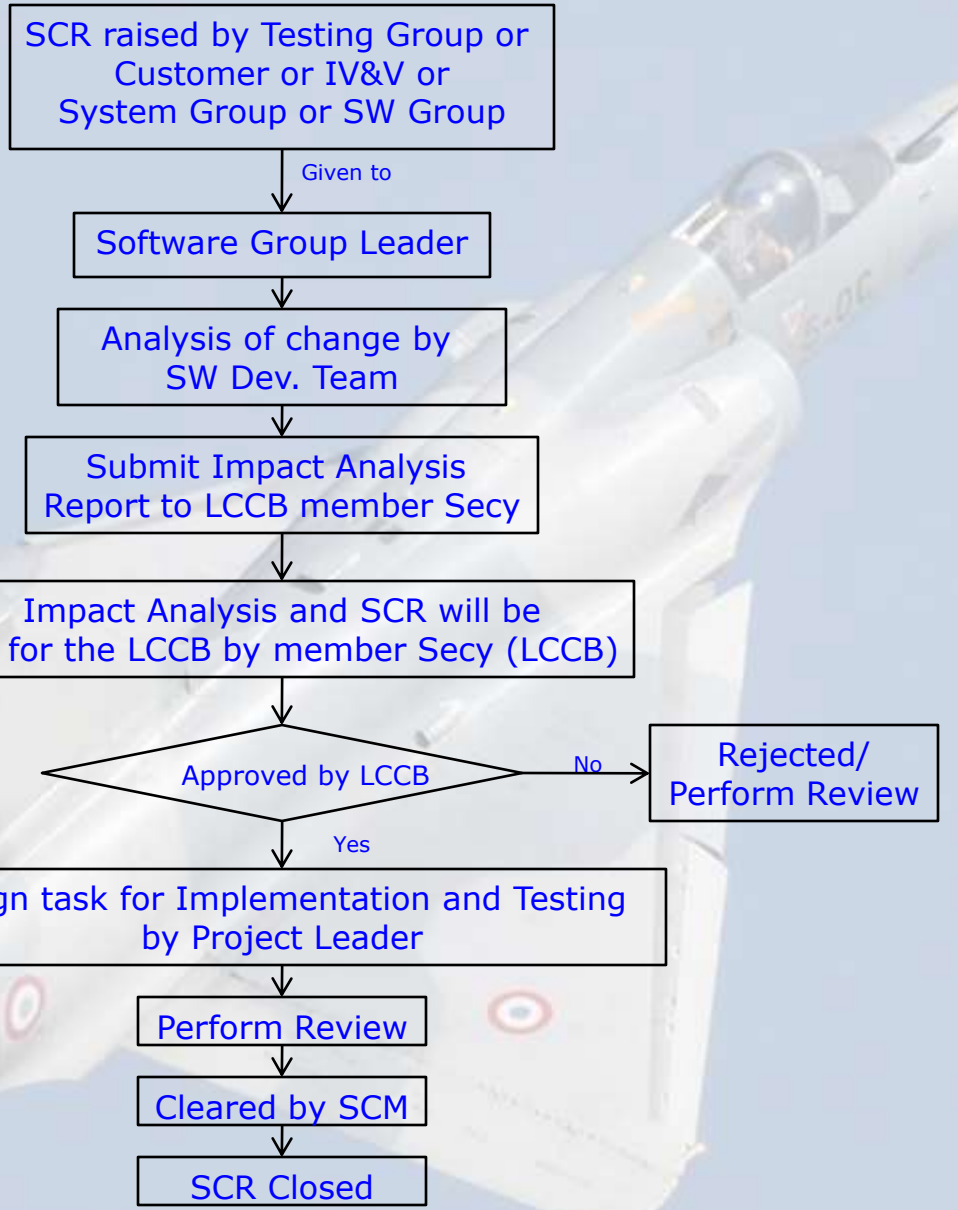
➢Change Management Process:

Objective of change Management is to provide for recording, evaluation, resolution and approval of changes throughout the software life cycle.

Once the baseline is created the changes may occur due to:

- ❖Change in requirement
- ❖Change due to integration tests / at different levels of Internal reviews
- ❖Changes due to incremental implementation of functionality.

Software Configuration Management Process



SOFTWARE CHANGE REQUEST			
Software Change Request No:			
Aircraft Name:			
Project/System Name:		Equipment / LRU Name:	NA
Request No (External Reference):			
Software Version No:		Equipment / LRU Mod Status:	NA
Change Type (Fill ( ) the appropriate one):			
New Requirement. Suggestion for Improvement. Design Change. Flight Test Feedback.		System Problem. Integration Problem. Hardware Change. Others. _____.	
References (e.g. Minutes of meeting, memos issues, technical notes generated etc):			
Notes: (If the change is necessitated due to any problem faced in the operational field units then complete the details of feedback / problem report from the services are to be given). NA			
Change Description:(Please attach supporting documents for the requested change)			
Originated By:			
Name: Designation: Date:			
Approved By (LCCB):			

	Project IV&V Member	System Group Leader	Sw Group Leader	Process Admin	Chairman
Signature: Name: Designation: Date:					

Remarks (if Rejected):	
Distribution:	**Closure Approval By (SCM):**
✓ SwGroup Leader. ✓ IV&V Team Leader. ✓ SPAG Team Leader.	Signature: Name: Designation: Date:

➢Re-Verification Activities:

Re-Verification activity is applicable if the changes are made to the final released baseline.

- ❖ Impact analysis, traceability matrix and changed artifacts will be the inputs.
- ❖ Re-Verification will be conducted to ensure that the changed software is error free and has not introduced any new errors in the already verified software.
- ❖ Depending on the extent of change, verification (Regression testing, Analysis or Reviews) is conducted to ensure that the software continues to perform as intended in the operational environment.

Certification Liaison Process

Purpose of Software Certification Liaison process is to establish communication and understanding between the applicant and the certification authority throughout the software life cycle to assist the certification process.

Activities:

- Submit Software Certification Plan and other requested data to the certification authority for review
- Resolve issues identified by the certification authority concerning the planning for the Software Certification.
- Obtain agreement with the certification authority on the Software Certification Plan
- Verification and validation for all Software artifacts of different phases of SDLC.
- Obtain the flight clearance certificate


THANK YOU