A-P-T RESEARCH, INC.

AN EMPLOYEE-OWNED COMPANY

MODEL BASED SYSTEM ENGINEERING & SOFTWARE SYSTEM SAFETY CONCEPTS, GOALS AND OBJECTIVES

A Presentation to G-48 MBSE and SSS Workshop
2-3 May 2017

Barry Hendrix
Sr. Principal Software Safety Engineer
APT Research, Inc.

MBSE AND SSS GOALS AND OBJECTIVES

- Understand MBSE Concepts on Modern Complex Programs
- Awareness of proliferation of MB Software Development
- Acceptance of various forms of MB by different agencies
- Ensure MBSE, MBSwEng, & Software /System Safety Integration Needs are Address
- Collaborate on Model Based Development and software system safety policies and best practices for Complex and Critical Systems

MBSE AND SOFTWARE SYSTEM SAFETY

- INCOSE, DOD, NASA endorsing MBSE as the norm on some complex and safety-critical systems
- DoD acquisition in place for models with software intensive with complex software functionality
- MBSE has engineering development advantages for software intense integrated systems
- MBSE used for interoperability and optimization to enhance performance in future battlefield scenarios.
- Model Based Development has emerged over the past decade as one solid solution proven to dovetail well with software engineering/software system safety goals and objectives

SYSTEM SAFETY…WHAT IS CURRENT AND NEXT?

- System Safety in past 50+ years has evolved from:
  - basic hazard analyses of federated hardware with much operator intervention to
  - highly integrated and complex software intensive SOS to
  - Sensor fused software systems of autonomy for command, control of complex interactions in safety-critical functions:
- Therefore System Safety programs and engineers must adapt to changing technology.
- Some History of System Safety Paradigms (DoD, FAA, Commercial)
  - 1967 MIL-STD-882A Hazard-based/Risk-Based
  - 1984 MIL-STD-882B Software Safety Analyses 300 Series Tasks Addition
  - 1994 IEEE STD-1228-1994 (MIL-STD 300 series software tasks commercialized) (G-48 Driven)
  - 1996 SAE ARPs 4761/4754 Functional-Based (FHAs), Criteria-based (DALS) for Military AC
  - 2008 LOR (AOP-52 US Navy, AM-COM 385-17 US Army, FMETs/FITs for USAF)
  - 2010 DO-331 Model Based Development & Verification (supplement to DO-178C)
  - 2010 ANSI - 010 Commercial System Safety (G-48 Driven)
  - 2012 MIL-STD-882E added FHAs and SOS Analysis (G-48 Influenced)
  - 2013 – 2015 Safety Case Initiatives/Workshop (G-48 Driven)
  - **2015 – 2017 Model Based Process Integration Initatives/Workshop (G-48 Driven)**

MBSE HIGHLY SUITED FOR COMPLEX SYSTEM OF SYSTEMS

- Times and technology are changing and so must MBSE and associated software system safety methods, processes.
- Real World Case Study Example:
  - Tri-Service F-35 JSF (Lightning) II is the most expensive and most complex software intensive safety-critical system in development.
    - Model Based on Vehicle Management Computers, C++ intensive, replaced Ada for safety-critical, Many SCF in SW are Model Based. Formal Methods, Goal Structuring Notation used to verify the most SCF implemented in software. Technical Integrity Required. Model Based Proven, Validated and Verified. Led to Airworthiness Certification and Customer Acceptance. Zero Mishaps.

SOME CURRENT PROGRESS IN MBSE AT RSA

- US Army Systems of Systems Emerging Well
  - Integrated Fire Protection Capability (IFPC) and Multi-Mission Launcher (MML) uses MBSE. Safety Significant requirements and SCF identified as part of the process.
  - SCF and process flow and behavior can be depiction using Magic Draw.
  - AGILE, SCUM used, but safety not compromised when implemented correctly.
  - FPGAs provide SCFs. A challenge to some and a new way of verifying safety…complex electronic hardware or software domain debates depending on programmable perspective and mental model. Model based being integrated into process.

- Candidate Systems for some MBSE include next Software Build of Integrated Battlefield Command System (IBCS). Initiatives underway to make that happen.

EVOLVING/EMERGING SOS AND MBSE, MBSS, SAFETY CASES

- The future will be more complex, more software model based and will require better methods for capturing objective safety evidence:
  - More C4IRS and C2 for US Army
  - More Autonomous Systems
  - Cyber, IA, Protection Technologies in Software (beyond Software Safety)
  - More Tri-Services coordinating for the current and future battlefield
  - NAVAIR, NAVSEA, NOSSA AOP-52 in place for Weapons with Computing.
  - F-35 progressing well as first big Model Based/Software Intensive System.
  - Navy UAVs in the future. Some Model Based Plans
  - USAF Long Range Strike Bomber awarded in 2016.
  - DoD/NASA/FAA are forward thinking and all have MBSE programs

SYSTEM SAFETY CULTURE MUST CHANGE WITH TECHNOLOGY

- Traditional system safety may not be adequate for emerging and evolving SOS and paradigm shifts…different “Mental Models” needed to adapt
  - JOINT, Multiple Contactors/Agencies, National Teams – Meshing Cultures/Methods
  - Highly Integrated, Sensor Fused, High Complexity, Various Levels of Abstraction
  - Proliferation of UAVs/AUS and autonomous systems, even driverless cars and robots.
  - Software Intensive with many Complex Interactions require new software safety methods
  - Model Based System Engineering (MBSE) is becoming the Norm on C2 and other emerging Information Centric SOS
  - Agile / Lean software development and less formal methods must be geared to include software safety more so than in past when used in rapid prototyping
  - Safety Cases WILL be needed to collect all aspects and objective safety evidence

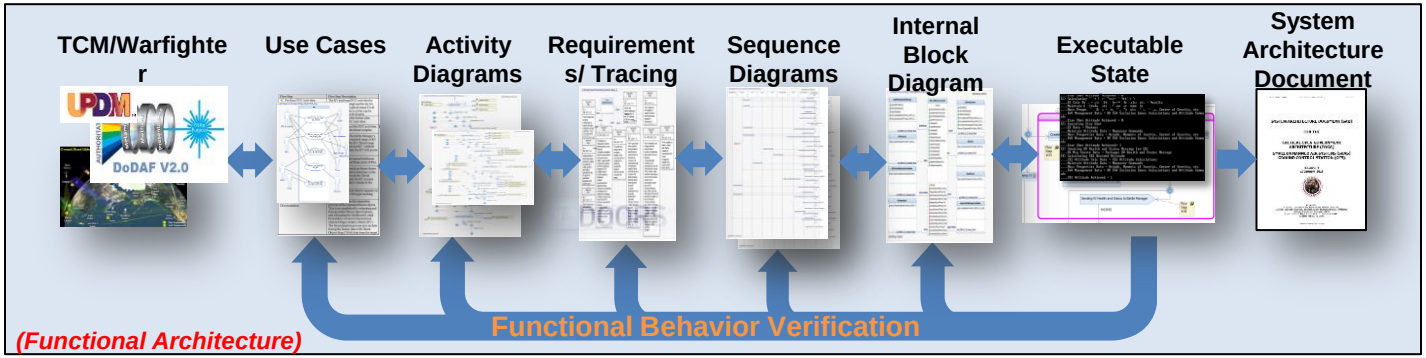
MODEL BASED SOFTWARE SYSTEM SAFETY

- Model Based Systems Engineering (MBSE) and Model Based Software development requires a different software system safety concept, application and effort because of :
  - Less traditional artifacts centric, less English Prose, More Graphical/Visual Notations
  - Need to show Functional and Physical Architecture, Integration and Interface
  - UML and SysML languages, System Architecturally Based, Model Must be Validated
  - More Models (Functions, Use Cases, Structural Diagrams, Activity Diagrams, Sequence Diagrams, Executable States Charts, Architectural System, Functional /Behavioral Verification, Complete Requirements Traceability, Handoff to Defined Systems
  - Auto-code Generation, Greater Level of Abstractions, Traditional CM must change
  - Complexity has outstripped human interpretation – models must capture attributes
  - Extracting Objective Evidence Needed for a Safety Cases

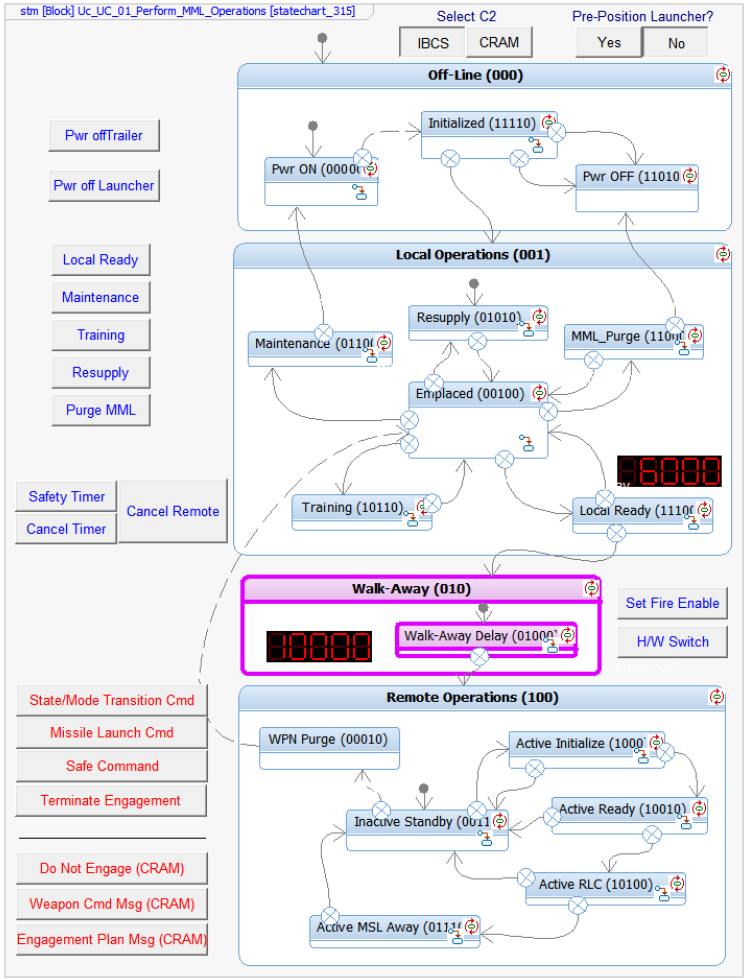
MBSE OUTPUT USEFUL FOR A SAR OR SAFETY CASE

Example of MBSE in place today by US Army AMRDEC Software Engineering Directorate CoMBAT Team Process

DoDAF MBSE Framework allows better Behavioral Allocations to be handed off to Software Developers (and Software Safety). Software Safety can have INCLUSIVE inputs at Use Case (beginning of the process)



MBSE – TYPICAL EXECUTABLE STATE BEHAVIORAL DIAGRAM (SAFETY FUNCTIONS CAN BE INTUITIVE)



MODEL BASED TOOLS

- Rhapsody provides a consistent design model that is also tied to requirements
- Rhapsody is one of many UML/SysML tools.
- Any programming language can be used for the actual embedded application software.
- Safety Attributes can be added throughout model based tools and linked to DOORS (to Rhapsody) or other safety requirements for full traceability from USE CASES, to DOORS, TO MODELING TOOLS and links to safety actions and produced artifacts.
- Many tools available with excellent capability to help augment Safety Case with graphical and highly visual flow representation of safety aspects and attributes vs. just words.
  - Rhapsody, SIMULINK, MATLAB, SCADE.

SAFETY EVIDENCE OBTAINED FROM MODELS

- Methods are evolving, but level of abstraction requires system safety to ensure objective safety evidence is obtainable from the Model
  - Ensuring software safety analysis of each Use Case for Safety-critical Functional; behavior – labor intensive as inputs to DOORs
  - Ensuring safety-significant attributes are flagged, tagged throughout model means software safety must keep up with evolving model as every stage
  - Use Cases, S-C Functionality, S-C requirements Assessment, safety test case allocation, safety verification and yielding OBJECTIVE SAFETY EVIDENCE (and Hazard Closure Evidence).
  - LOR, DA and especially Special Safety Tests can be modeled more precisely in models:
    - Off Nominal Tests (ONT), Failure Modes Effects Tests (FMETs), Failure Immunity Tests (FITs) and unlimited fault insertion to determine FAILURE CONDITION behavior response and to refute safety arguments to validate Safety Cases.
  - Safety Cases with Evidence or Refuted Arguments will be required as current SARs are inadequate as current written for Model Based Software System Safety

SOME SYSTEM SAFETY ADVANTAGES OF MODEL BASED DEVELOPMENT

- If so designed MBSE can show the “big safety picture” and explicit safety functions, safeguards, safety features with easy to interpret sequence flow diagrams and behavioral flow diagrams of safety-critical functions.
- MBSE improves engineering collaboration, teaming and communications across domains – same core representation – for safety documentation
- System engineering, software engineering and safety engineering processes and actual FUNCTIONS and Normal/failure CONDITIONS can be visualized vs. word interpretation that can be vague and ambiguous
- Proposed changes (safety changes) can be evaluated
- More consistent safety documentation, traceability, improves technical integrity
- Already validated auto-code generation using the tools to perform them can be better analyzed in a model based setting. A plus for safety.

CLOSING THOUGHTS AND RECOMMENDATIONS

- MBSE, MBSwEng, Software System Safety must be integrated into a “Golden Triangle” for Success
  - DoD with help of INCOSE and large Prime contractors is in transition to current and emerging engineering methods to keep from falling behind
  - Software safety involvement and contributions require open mindedness and transition from older traditional methods. Times, technology and environments are changing and system safety and software safety must adapt and help make it all work.
  - Cultural changes needed, management buy-in needed. Convincing ourselves better evidence based safety is needed and emerging methods will really work.
  - Any safety-critical program with MBSE, MBD must ensure an adequate SSPP and SWSSPP (WHAT) with subordinate processes (HOW) are developed.
  - Flexible Policies, Best Practices, Processes Needed for Integrating Model Based aspects of Systems Engineering, Software Engineering, System Safety/Software System Safety

SOME LEADING EDGE ENDORSEMENTS FOR MBSE

Model Based System Engineering – INCOSE, UK, Education Academia
http://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf?sfvrsn=0

https://incoseonline.org.uk/Documents/zGuides/Z9_model_based_WEB.pdf

http://syse.pdx.edu/program/portfolios/julia/MBSE.pdf

DoD Acquisition: Model Based Systems Engineering Development Briefs
http://www.acq.osd.mil/se/briefs/2014_04_03-Zimmerman-SEDC-Final.pdf

Recent Links on how some Army programs are progressing well with Agile and Lean Software:
http://usaasc.armyalt.com/#folio=60 Page 59, Agile Acquisition, Ranjot Singh Mann P.E., and Michael Hanners

Model Based Safety Analysis by NASA and NASA Langley
http://shemesh.larc.nasa.gov/fm/papers/Model-BasedSafetyAnalysis.pdf

BACKUP INFO – DO-331 MBD OVERVIEW (COMMERCIAL/CIVIL/CERTAIN MILITARY AIRCRAFT)

DO-331 Model Based Development and Verification Supplement to DO-178C and DO-278A

L. Alford

APT Research, Inc

A-P-T RESEARCH, INC.

AN EMPLOYEE-OWNED COMPANY

DO-331 MODEL BASED DEVELOPMENT AND VERIFICATION SUPPLEMENT TO DO-178C AND DO-278A

L. Alford

APT Research, Inc

OBJECTIVES

- **Objectives for DO-178C suite of documents, including the Supplements:**
  - Promote safe implementation of aeronautical software
  - Provide clear and consistent ties with the systems and safety processes
  - Address emerging software trends and technologies
  - Implement an approach that can change with the technology
  - Industry-accepted guidance for satisfying airworthiness requirements for avionics equipment

19

PURPOSE

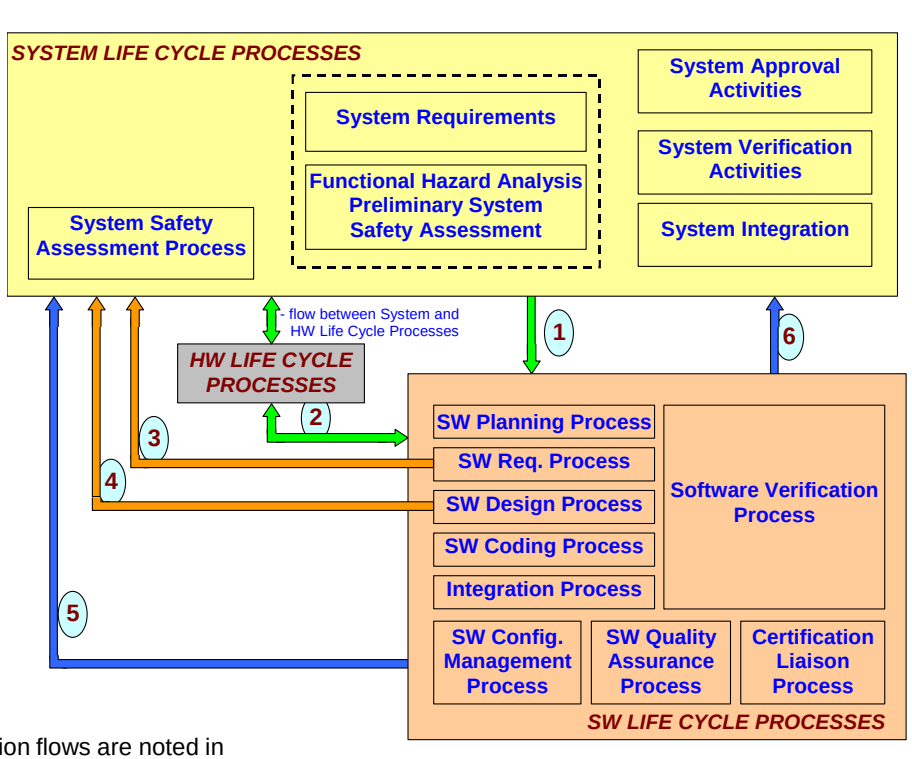
- **Industry-accepted guidance for satisfying airworthiness requirements for avionics equipment**
  - To provide guidelines for software to comply with
    - Proof of no intended function
    - Proof of performance in an avionics LRU installation
  - To provide agreed criteria consistent with civil certification authorities
  - By treaty agreement, this applies to NATO nations and any other countries recognizing this set of guidelines for aviation software

- **Results Needed**
  - Agreed criteria for airworthiness certification requirements for software that doesn't differ from one person or certification authority to another
  - Allows for recognition of an aircraft model capability by air traffic control for airspace access and interoperability
    - This last is an issue for all military aircraft

20

Information flow between System & SW life cycle processes

Context for use of DO-331 MBD



ARP4754A

DO254

DO178C

DO-331* is used with DO-178C

*It adds and modifies DO-178C objectives for MBD aspects

More detailed information flows are noted in backup charts

21

- **Its about identifying the "safe-subset" use of MBD technology to be used in safety related applications**
  - Same role as the suite of DO-178C documents
  - It applies "error class analysis" to determine *what needs to be considered* for MBD projects to confirm best known practices and proof of safety
- **Its about using suitable graphical engineering methods to design a software system**
  - The ability create graphic representations of requirements, architecture and designs has existed for some time
  - Visual format promotes better understanding of the system and its interactions
  - The use of graphics has been refined with semantics of notations with more rigorous syntax and less ambiguity – leading to the use of analysis techniques on models within the modelling environment to remove errors early in the lifecycle

DO-331 MBD FUNDAMENTALS - 2

- **Clear distinctions are made between 2 types of graphical models:**
  - Specification Models – Defining high level requirements without implementation, software architecture, or data flow and/or control flow
  - Design Models – Defining architecture and design (low level requirements)
    - If code can be written from the model, then it is considered a Design Model
    - A Design Model must have parent requirements in scope of the DO-178C development process
  - Note that Systems Engineering may be the author of a Specification Model and therefore subject to meeting the objectives of DO-331 for that model

DO-331 MBD FUNDAMENTALS - 3

- **Determining which artifacts will be in a model drives the determination of applicable objectives and activities**
  - If the model is defining requirements without indicating how it will be accomplished, then the Software Requirements Document (SRD) becomes the location for that model
  - Detailed architecture, data and control flow, implementation and performance form the content of the Software Design Document (SDD)
  - A MBD area of a system will continue to include:
    - Full requirement traceability and model traceability
    - Configuration control including the models and elements used
    - Verification of the models, libraries, and model elements

Model Usage Examples

1. Example 1: Simplest and common use of MBD; the Design Model goes into the Software Design Document (SDD)
2. Example 2: the Design Model is developed from the requirements contained in the Specification Model.
3. Example 3: the textual description refers to LLR and possibly architecture: DO-178C guidance is applicable to these.
4. In examples 4 and 5, separating system and software life cycle data may be difficult: the artifacts may serve for both the systems and software groups. Use the guidance in DO-331 as the compliance criteria for the artifact(s). The MBD guidance for HLR applies to system and software Specification Models, while the MBD guidance for Low-Level Requirements (LLR) applies to software Design Models.
5. Examples 6 and 7 are evolving now and are added to provide guidance. These are not currently represented in DO-331, though the planning, activities, artifacts, and relationships are defined in DO-331.

Initiating Process for Life Cycle Data	Model Example 1	Model Example 2	Model Example 3	Model Example 4[1]	Model Example 5[1]	Model Example 6[4]	Model Example 7[4]
System Requirements Process	Textual source requirements allocated to software	Textual source requirements for the model(s)	Textual source requirements for the model(s)	Textual source requirements for the model(s)	Textual source requirements for the model(s)	Textual source requirements for the models	Specification Model(s)
System Design Process					Design Model(s)	Specification Model(s)	Textual source requirements for the model(s)
Software Requirements Process	Textual source (SRS) requirements for the model(s)	Specification Model(s)[2]	Specification Model(s)	Design Model(s)		Design Model(s)	Model examples 1 - 5
Software Design Process	Design Model(s)	Design Model(s)	Textual Design Description (SDD)[3]				
Software Coding Process	Source Code	Source Code	Source Code	Source Code	Source Code	Source Code	Source Code

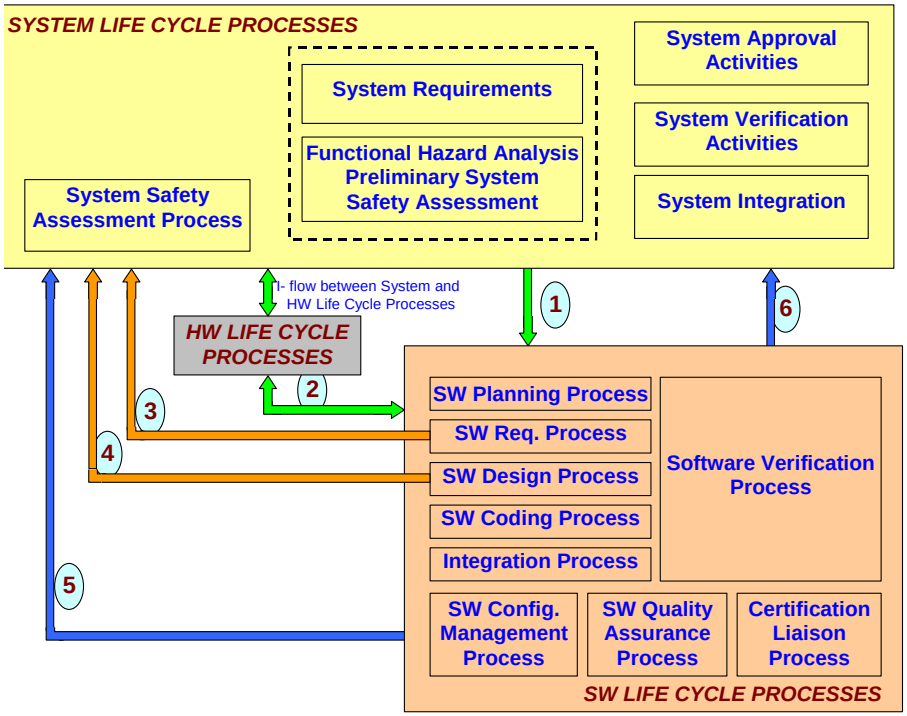
DO-331 MBD FUNDAMENTALS - 4

- **MBD Data Items (beyond the normal items) to be expected in a program:**
  - Model Planning
    - How it will be used and how and where it fits into the lifecycle; what Model Standards will be used; the verification approach; simulation - if used for credit
  - Model Standards and Techniques
    - The guides for both Specification and Design models, including constraints, instructions, language, symbols used, model element libraries
  - Model Element Libraries
    - Each element must be assured to meet the required Software Level as it is a set of executable code that generates a symbol and associated action. A full data package for each library is necessary
    - Unused elements should be removed from the library, unless the standard includes instructions prohibiting use, particularly for unassured elements

DO-331 MBD FUNDAMENTALS - 5

- **MBD Data Items to be expected in a program, continued:**
  - Model Coverage
    - Analysis which identifies requirements in a Design Model not verified by requirements testing;
    - This may identify unintended functionality
    - Criteria for this analysis and resolution of issues found must be defined in the planning document
  - Model Simulation
    - This activity exercises the model behavior using a simulator
    - If used for credit, the simulation cases, procedures and results are necessary

Information flow between System & SW life cycle processes



1

System process shall give the following data to SW process

⇒ **To planning process**

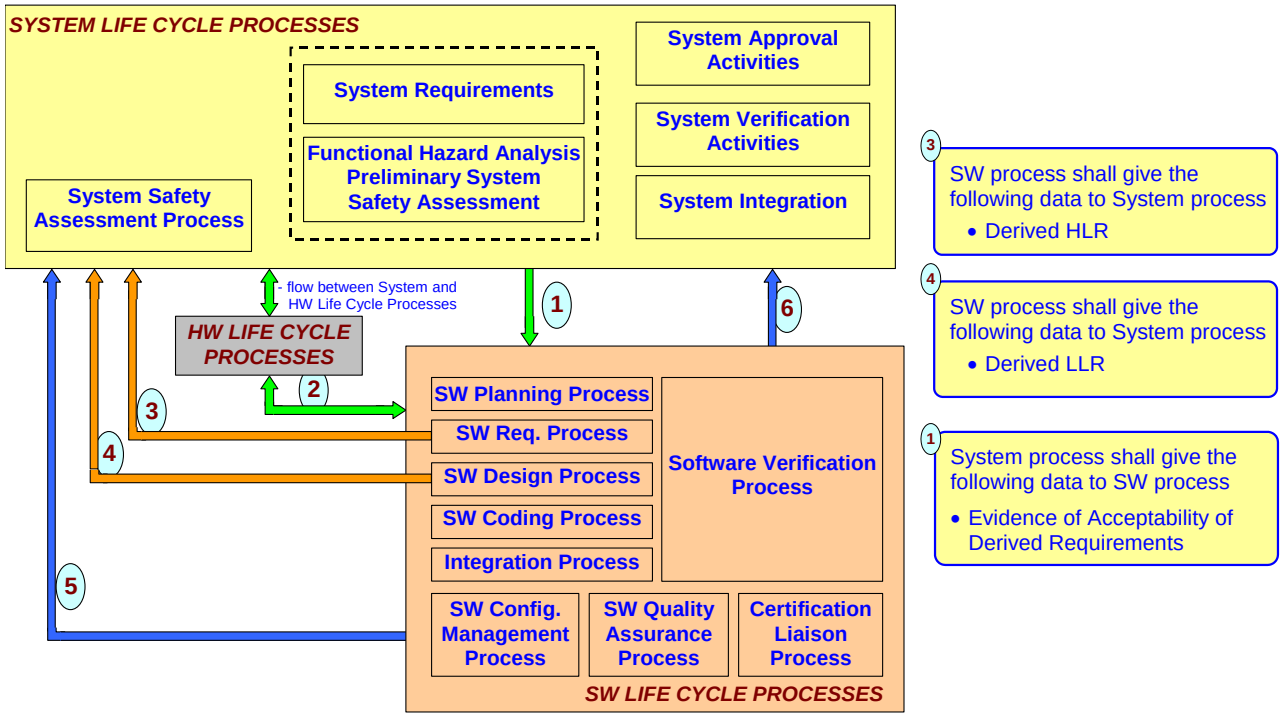
- System Safety Objectives
- Software Level(s)
- System Description and Hardware Definition

⇒ **To development process**

- System Requirements Allocated to Software
- Design Constraints

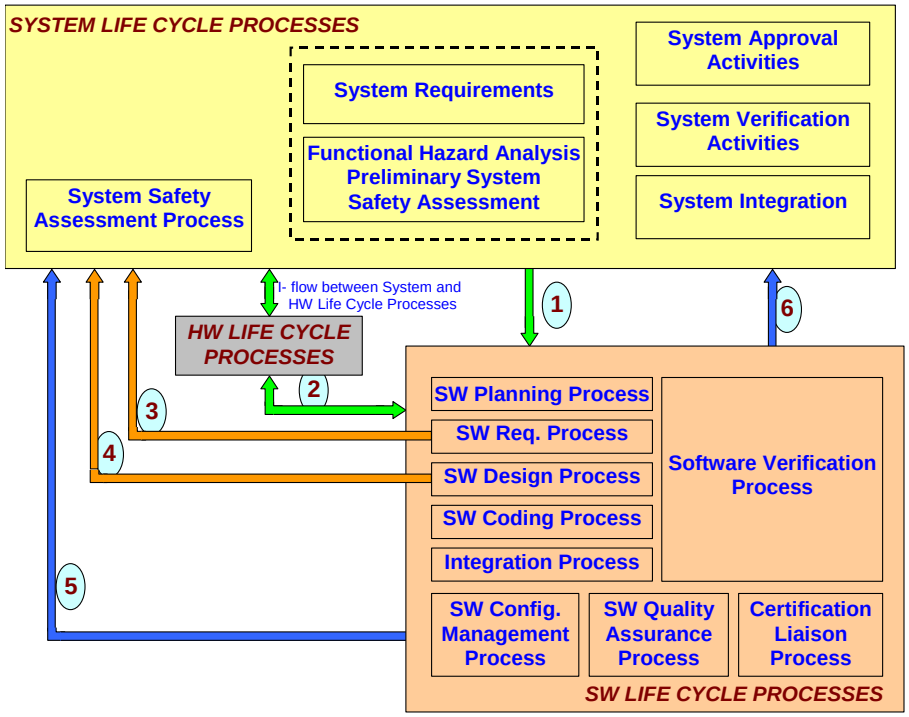
The data flow between systems, software and hardware are critical to success and should be confirmed

Information flow between System & SW life cycle processes



The data flow between systems, software and hardware are critical to success and should be confirmed

Information flow between System & SW life cycle processes



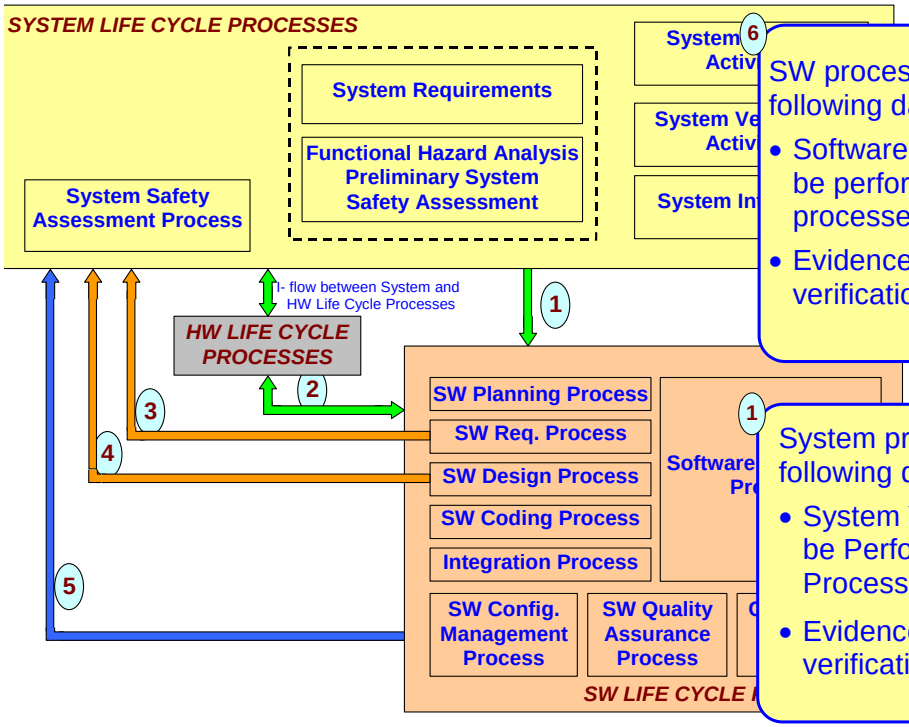
5 SW process shall give the following data to System process
- Problem or change documentation

6 SW process shall give the following data to System process
- Any limitation of use
- Configuration identification data

1 System process shall give the following data to SW process
- Evidence of Acceptability of Problem or change documentation

The data flow between systems, software and hardware are critical to success and should be confirmed

Information flow between System & SW life cycle processes



SW process shall give the following data to System process

- Software verification activities to be performed by system processes
- Evidence of any system verification

System process shall give the following data to SW process

- System Verification Activities to be Performed by Software Processes
- Evidence of any SW verification activities performed

The data flow between systems, software and hardware are critical to success and should be confirmed