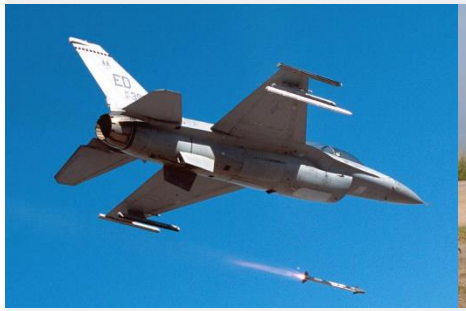
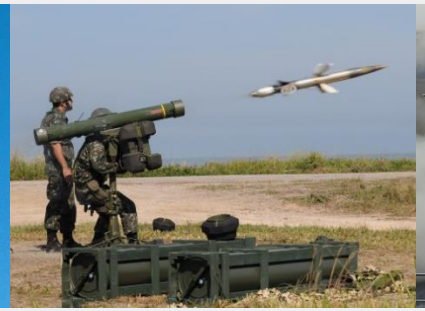
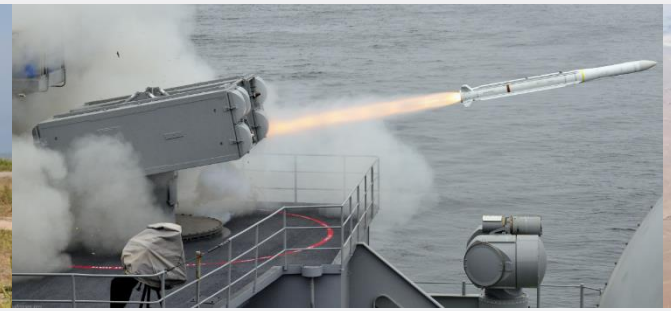
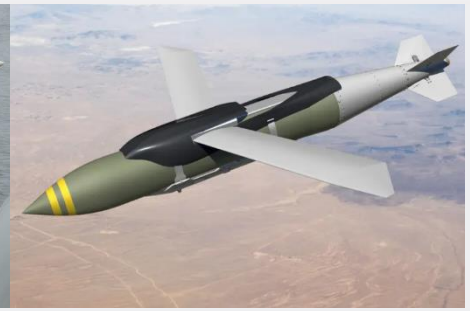
Australian Government
Department of Defence
Capability Acquisition and Sustainment Group

Software Safety Assurance for Explosive Ordnance

Mr Warren Miller
Software Assurance Desk Officer

Directorate of Engineering
Explosive Materiel Branch (EMB)

Defending Australia and its National Interests
www.defence.gov.au

Foreword

- A Paper with the same title as this presentation has been produced and will be published as part of the Safety Symposium proceedings.

- Additional information is provided in the Paper to that presented here.

Agenda

- Definitions
- Software in Munitions
- Guided Missile Example
  - Subsystems
  - Theory of Operation
  - Software Functions
  - Software Contribution to Hazardous System Conditions
- Minimising Software Contribution to Hazardous Conditions
- Software Safety Assurance
  - Objective Quality Evidence (OQE) Requirements
  - Bespoke vs Off The Shelf vs Armaments Cooperative Program

What is meant by the term “Software”:

- Catch-all umbrella term for where logic is programmed into the EO
  - Software programs
  - Settings / database / configuration files
  - Firmware
  - Software-like devices
    - programmable logic devices and
    - similar hardware

Definitions

What is meant by the terms “Systems Engineering” and “Software Engineering”

- Systems Engineering
  - Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.

- Software Engineering
  - The application of a systematic, disciplined approach for the development, operation, maintenance and retirement of software.
  - The process of analysing user requirements and then designing, building and testing a software application which will satisfy those requirements

What is meant by the terms “System Safety Engineering” and “Hazard”

- System Safety Engineering
  - The application of engineering and management principles, criteria, and techniques to achieve acceptable risk, within the constraints of operational effectiveness and suitability, time, and cost, throughout all phases of the system life cycle.
  - Focuses on identifying hazards, their causal factors, and predicting the resultant severity and probability. The ultimate goal of the process is to eliminate or reduce the severity and probability of the identified hazards, and to minimize risk and severity where the hazards cannot be eliminated.

- Hazard
  - A real or potential condition that could lead to an unplanned event or series of events (i.e. mishap) resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment.

What is meant by the terms “Software Assurance” and “Software Safety Assurance”

- Software Assurance
  - Software Assurance is defined as the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software functions in an intended manner.

- EO Software Safety Assurance
  - Software Safety Assurance within the EO domain can be defined as the level of confidence that munitions software will execute within its system context and operational environment with an acceptable level of safety risk.

EO hosted Software:

- AGM-154 C BLOCK II Joint Stand-off Weapon (JSOW)
- AGM-88C High-Speed Anti-Radiation Missile (HARM) – BLK V Software
- AGM-88E Advanced Anti-Radiation Guided Missile (AARGM) – BLK1 MT2
- AIM-9X Sidewinder – OFS 9.150 + IMU, CAS, Cryo Software
- AMRAAM AIM-120D – Software Improvement Program Load 2 (SIP2) Tape 24 Revision 22
- GBU-39 (SERIES) Small Diameter Bomb – Mission Computer/OFP
- GM-84 Harpoon Missile variants
- Joint Direct Attack Munition (JDAM)
- RBS 70 BOLIDE MK C
- RIM 162 Evolved Sea Sparrow Missile (ESSM)
- RIM-66M-9 and RIM-66M-10 Standard Missile 2 (SM-2)

EO interfacing Software:

- Common Munitions BIT/Reprogramming Equipment Plus (CMBRE+) Munitions Application Program (MAP) Software
  - Runs and reports on Built In Test for supported munitions
  - Installing new software loads to supported munitions

- SM-2 MK 698 Guided Missile Test Set (GMTS)
  - All Up Round (AUR) operational checkout
  - AUR preventive maintenance, and rectification and verification testing
  - Validation check of the missiles software suite to verify the validity of software configuration

Example: Generic Guided Missile – Precision Guided Munition (PGM)

- Thermal batteries;
- Propulsion system;
- Navigation system;
- Guidance control;
- Tactical Data Link (TDL);
- Target sensor;
- Programmable fuze;
- Proximity sensor target detector;
- Contact sensor;
- Warhead; and
- Multiple distributed Computer Software Configuration Items (CSCIs)

Sequence of Missile Operating Modes

- Powered Off Mode
- Powered On Mode
- Launch Mode
- Separation Mode
- Fly Out Mode
- Detonation Mode

Other Modes

- Training Mode
- Test Mode
- Abort Fly Out Mode

Generic PGM Example – Theory of Operation including Software Functions

Sequence of Missile Operating Modes with Software Functions highlighted in blue

- Missile Powered Off Mode
  - Missile loaded onto LP
- Missile Powered On Mode
  - LP power applied
  - Missile Power on Built in Test (PBIT) completed and status reported to LP
    - Perform PBIT
    - Send PBIT status to LP
  - Missile Cyclical Continuous BIT (CBIT) and status reporting initiated
    - Perform CBIT
    - Send CBIT status to LP
  - LP Operator sets SAFE-ARM switch to ARM
  - LP Operator depresses and releases Missile LAUNCH button

Generic PGM Example – Theory of Operation including Software Functions

- Missile Launch Mode
  - Missile pre-launch functions initiated
    - Thermal batteries activated
      - Activate GCS Battery & Activate Target Sensors, Navigation and Guidance Battery
    - Fins deployed, unlocked and tested
      - Unlock Guidance Fins & Perform Wiggle Test
    - Wings unlocked to position one, partially extended
      - Deploy wings to position one, partially extended
    - Sensors startup
      - Start IR Sensor & Start Active Radar Sensor
    - GPS startup
      - Acquire GPS Satellite Connection

Generic PGM Example – Theory of Operation including Software Functions

- Missile Launch Mode contd
  - LP downloads data to missile
    - Accept LP, target and flight plan data from LP
  - LP removes power from the missile and retracts umbilical cable
    - Detect umbilical cable disconnect
  - Missile initialises and optimises flight and detonation parameters using LP data
    - Initialise and optimise flight parameters using LP data
    - Initialise and optimise detonation parameters using LP data
  - Missile pre-launch functions completed
  - Missile sends ready to launch signal to LP launcher
    - Send ready to launch signal to LP

Generic PGM Example – Theory of Operation including Software Functions

- Missile Separation Mode
  - Missile rocket motor ignites
    - Launch Weapon (ignite rocket motor)
  - Missile launches
  - Missile clears rail
  - Missile attitude holds state until safe separation distance from the LP is achieved
    - Hold Attitude State

Generic PGM Example – Theory of Operation including Software Functions

- Missile Fly Out Mode
  - Missile warhead armed when safe separation distance from the LP is achieved.
    - Arm warhead & Initialise fuze parameters
  - Missile navigation and guidance systems control the missile flight path, guiding the missile to the target coordinates via any waypoints
    - Get GPS guidance information
    - Get IMU guidance information
    - Dead reckon target between mid-course updates
    - Wings are unlocked to position two, fully extended, if extended range required
      - Deploy wings to position two, fully extended
    - Missile accepts mid-course updates via TDL from the LP or other controlling unit if weapon handed off
      - Accept mid-course updates
    - If command received to abort, missile guided to safe location and self-destructs
      - Abort fly-out

Generic PGM Example – Theory of Operation including Software Functions

- Missile Fly Out Mode contd
  - Missile navigation and guidance systems control the missile flight path, guiding the missile to the target coordinates via any waypoints contd
    - Weapon guidance algorithms will automatically avoid friendly and no-strike forces
      - Guide to target (avoid friendly and no-strike forces)
    - During rocket motor burn, the missile guidance system controls the missile flight path using a combination of fin deflections and JVC vectored trust.
      - Guide to target using combination of fin deflections and JVC vectored trust (during rocket motor burn).
    - After rocket motor burnout, the autopilot controls missile flight path using only fin deflections.
      - Guide to target using fin deflections only (after rocket motor burnout).

Generic PGM Example – Theory of Operation including Software Functions

- Missile Fly Out Mode contd
  - Missile autonomously acquires the target using its on-board sensors and engages the target independently
    - Acquire target using IR Sensor
    - Acquire target using active Radar sensor
  - Missile guidance system controls the missile flight path, guiding the missile to the target
    - Proximity sensor senses target
      - Sense target using proximity sensor
    - Missile transmits a Bomb Impact Assessment (BIA) to the TDL network immediately prior to impact
      - Transmit Bomb Impact Assessment (BIA) (prior to impact)

- Missile Detonation Mode
  - Missile detonates prior to, at or after impact based on target composition
    - Detonate warhead

Software Contribution to System Hazards

- Software, though itself not inherently dangerous, through its system control and monitoring functions may be a causal or contributing factor in transitioning explosive ordnance hardware into an unsafe state.

- Hazardous software behaviour could result from either:
  - unanticipated behaviours and interactions arising from software design decisions; or
  - errors introduced during the software development process

Generic PGM Example –Software Contribution to System Hazards

ID	Software Function	Hazard	Hazardous Event / Mishap
1	Activate Battery	Inadvertent battery activation	Personnel come in contact with hot battery or missile section.
2	Launch Weapon	Inadvertent Weapon Ejection	Weapon impacts location other than the target
3	Launch Weapon	Failure to eject	Personnel come in contact with hot hung store.
4	Detect umbilical cable disconnect	Failure to eject	Personnel come in contact with hot hung store.
5	Deploy wings	Inadvertent Wing Deployment to position 2 (captive)	During carriage could result in weapon contact with aircraft or adjacent weapon
6	Deploy wings	Wings do not deploy to position 1 (loss of stability)	Weapon impacts location other than the target
7	Deploy wings	Wings do not deploy to position 2 (loss of glide capability)	Weapon impacts location other than the target
8	Deploy guidance fins	Fins do not deploy (1 or more fins)	Weapon guidance will fail resulting in uncontrolled flight.
9	Acquire GPS satellite connection	No GPS guidance	If the weapon is not able to acquire satellite connection, the weapon will still guide however the CEP would be increased by an unknown amount (the IMU drift rate is not known).
10	Get GPS/IMU guidance information	Loss of weapon guidance/control	If GPS provides incorrect guidance information the weapon will impact a location other than the target.
11	Control guidance fins	Fin deflection at launch	Aircraft recontact. The worst case consequence is aircraft recontact resulting from full fin deflection at launch.
12	Arm Fuze	Unintended fuze arming	Fuze could arm prior to achieving safe escape from LP
13	Arm Fuze	Failure to arm / fire	Weapon will not detonate on impact with target

Minimising Software Contribution

- Many EO software functions have the potential to cause or contribute to hazardous conditions.
- Given this is the case, safety must be designed-in i.e. hazards designed-out, and any residual risk minimised.
- To do so requires a systematic and rigorous approach to software system development and system safety.
- Adopt a set of industry standards and guide books to provide a systematic approach.
- Apply level of rigour commensurate with software contribution to system safety risk.
- Ensure the Principles of Software Safety Assurance are satisfied.

Lifecycle Standards

- System Lifecycle Standard
  - ISO/IEC/IEEE 15288:2015(E) Systems and software engineering — System life cycle processes
  - ISO/IEC/IEEE 15289:2015 Systems and software engineering — Content of life-cycle information products (documentation – systems engineering)
- Software Lifecycle Standard
  - Legacy Standard:
    - MIL-STD-498 Military Standard: Software Development and Documentation
  - Modern Standards:
    - ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes
    - ISO/IEC/IEEE 15289:2015 Systems and software engineering — Content of life-cycle information products (documentation – software engineering)

Safety Standards and Guides

- System Safety Standard
  - MIL-STD-882E System Safety Standard 11 May 2012

- Domain Specific Software Safety Engineering and Assurance Guide
  - AOP-52 (STANAG 4452) Guidance on Software Safety Design and Assessment of Munition-related Computing Systems, Edition B Version 1 – 29 November 2016

Scaled Level of Rigour

- Assessment of risk for software, can't rely on severity and probability. Determining the probability of failure is difficult at best.
- MIL-STD-882E and AOP-52 describe an alternative approach to assess software's contributions to system risk that considers the potential risk severity and the degree of control that software exercises over the hardware.
- The resultant Software Safety Criticality Index (SSCI) determines the level of rigour to be applied to design, coding, integration and testing activities. SSCI Matrix from AOP-52:

Hazard Severity / Software Control Category		Catastrophic	Critical	Marginal	Negligible
I	Autonomous	1	1	2	3
IIa/IIb	Semi-Autonomous	1	2	3	4
IIIa/IIIb	Redundant Backup	2	3	4	5
IV	Not Safety Related	3	4	5	5

Scaled Level of Rigour - Example

ID	Software Function	Hazard	Hazardous Event	Severity	Software Control Category	SSCI
1	Activate Battery	Inadvertent battery activation	Personnel come in contact with hot battery or missile section.	Negligible	IIIa	5
11	Control guidance fins	Fin deflection at launch	Aircraft recontact. The worst case consequence is aircraft recontact resulting from full fin deflection at launch.	Catastrophic (Aircraft Recontact)	I	1

Severity		
Catastrophic	1	Could result in one or more of the following: death, permanent total disability, irreversible significant environmental impact, or monetary loss equal to or exceeding $10M.
Negligible	4	Could result in one or more of the following: injury or occupational illness not resulting in a lost work day, minimal environmental impact, or monetary loss less than $100K.

Software Control Category	
I	Software exercises autonomous control over potentially hazardous hardware without the possibility of intervention to preclude the occurrence of a hazard. Failure of the software or a failure to prevent an event leads directly to a mishap.
IIIa	Software item issues commands over potentially hazardous hardware systems, subsystems or components requiring human action to complete the control function. There are several, redundant, independent safety measures for each hazardous event.

Severity and SCC definitions per AOP-52.

Principles of Software Safety Assurance

- These principles are evident in the standards and guides

- Principle 1: Software safety requirements shall be defined to address the software contribution to system hazards.
- Principle 2: The intent of the software safety requirements shall be maintained throughout requirements decomposition.
- Principle 3: Software safety requirements shall be satisfied.
- Principle 4: Hazardous behaviour of the software shall be identified and mitigated.

- The confidence established in addressing the software safety principles shall be commensurate to the contribution of the software to system risk.

Objective Quality Evidence (OQE)

- Software safety assurance assessment is heavily dependent on the quantity and quality of OQE produced by the munition's development program.
- Historically, the OQE varies considerably between programs. The specific supplier, the type of program and contracted list of deliverables are contributing factors.
- Broadly speaking, in the context of munitions, CASG oversees three types of development and procurement programs
  - Bespoke
  - Off the Shelf (OTS)
  - Armaments Cooperative Program (ACP)

Bespoke

- Bespoke software development project where CASG has considerable input and visibility into the software safety and software engineering processes. Software development may be performed internally within Defence but more often is an external procurement activity. Within this category, in future, EO will additionally be sourced under sovereign programs using the GWEO strategic partners.

- Software safety assurance activities for bespoke software may be performed progressively during the development program using a comprehensive set of OQE and involvement in document review/approval, design reviews, system/software safety workgroups and test/verification events. Software safety assurance staff will have the benefit of visibility additionally into ‘ensurance’ activities. With this combined high level of visibility, a robust software safety assurance assessment can be constructed. The number of engineering staff allocated to software safety assurance activities must be appropriate and commensurate with the size and complexity of the software program.

Bespoke contd

- Clear visibility into software system engineering and safety engineering activities, and their output products, via contractual clauses that are levied on the supplier driving:
  - process;
    - through standards;
  - deliverables;
    - through the Contractor Data Requirements List (CDRL); and
  - customer visibility;
    - through reviews – design reviews, safety working groups, anomaly review boards, etc.; and
    - through observation/witness of test and verification activities.

Of the Shelf (OTS) and Armaments Cooperative Program (ACP)

- Of the Shelf (OTS) procurement where the EO has been developed and introduced into service by a foreign nation and CASG has little ability to influence software development or limited ability to gain data on specifics of software development. Mechanisms included under this category include Foreign Military Sales (FMS) and Direct Commercial Contract (DCC).

- Armaments Cooperative Program (ACP) where there is joint Research, Development, Test and Evaluation (RDT&E), acquisition and production of defence technologies, systems or equipment with partner and allied nations. It is expected under this type of program that CASG would have some limited visibility into, and possibly participation in, the development program.

Of the Shelf (OTS) and Armaments Cooperative Program (ACP) cont

- For OTS/ACP software, it is expected that the assurance activities outlined in the bespoke approach were previously performed by the foreign developing organisation and certifying authorities. Software safety assurance activities are performed later in the product lifecycle, to a shallower depth using a more restricted set of OQE, with limited visibility into the development program. Certification by a competent foreign certifying agency may be leveraged to support the safety assessment. Differing end-use Configuration, Role and Environment (CRE) are factors that also must be taken into consideration during the assessment.

- Under ACP arrangement, ADF involvement will provide greater scope to access the desired data however intellectual property and/or export restrictions may still enforce limitations. Therefore, the OTS assurance approach will more likely be adopted than that for bespoke programs

OTS & ACP – What to do when minimal OQE has been furnished

- Document the assurance deficit and if possible quantify the risk;
- Use prior Certification (by competent & independent assessor) as a foundation. Examples:
  - US Air Force – Non-Nuclear Munitions Safety Board (NNMSB), USAF.
  - US Army – Army Weapon System Safety Review Board (AWSSRB), US Army.
  - US Navy – Naval Surface Warfare Centre (NSWC) – Corona Division
  - US Navy – Weapon System Explosive Safety Review Board (WSESRB) via the Software Systems Safety Technical Review Panel (SSSTRP), or the Fuze and Initiation System Technical Review Panel (FISTRP).
  - Defence Ordnance Safety Group (DOSG), United Kingdom.
- Request further OQE from the supplier;
- Conduct testing to generate desired OQE; or
- Combination of the above

OTS & ACP – Ensuring adequate OQE has been furnished

- Start early in the Acquisition process
- Communicate OQE requirements with potential suppliers during the tendering phase
- Negotiate a comprehensive considered data deliverables list with the selected supplier and embed it in the acquisition contract

Questions

- Any questions?