

Product Service History for the Qualification of Safety- Critical Ground Segments

Emmanuel Lesser | 09/24/2025

Tim Arulsuthan

RESTRICTION ON USE, PUBLICATION DISCLOSURE OF PROPRIETARY INFORMATION AND IMAGES

This document contains information proprietary to MDA Space Ltd. ("MDA Space"), its subsidiaries, affiliates or to a third party to whom MDA Space may have a legal obligation to protect such information from unauthorized transfer, export, use, reproduction, or duplication. Any disclosure, transfer, export, use, reproduction, or duplication of this document, or of any of the information or images herein, other than for the specific purpose for which it was disclosed is expressly prohibited, except as MDA Space expressly agrees to in writing.

© 2025 MDA Space, subject to General Acknowledgements for the third parties whose images have been used in permissible forms. All rights reserved.



MDA SPACE OVERVIEW

55-year history of space innovation

4,000+ global workforce

500,000+ sq. ft. of design, manufacturing & testing facilities

3 Business Areas – Robotics & Space Operations, Satellite Systems, & Geointelligence



WE DEVELOP ADVANCED SPACE TECHNOLOGIES THAT ENABLE MISSION FIRSTS

Sapphire mission for space
domain awareness

Three generations of
RADARSAT satellites
operating since 1995

MDA robotics on world's first autonomous
on-orbit servicing mission

MDA sensors have been
operational on 15+ Cygnus
missions

MDA sensors and robotics
have been operational on
Mars since 2008

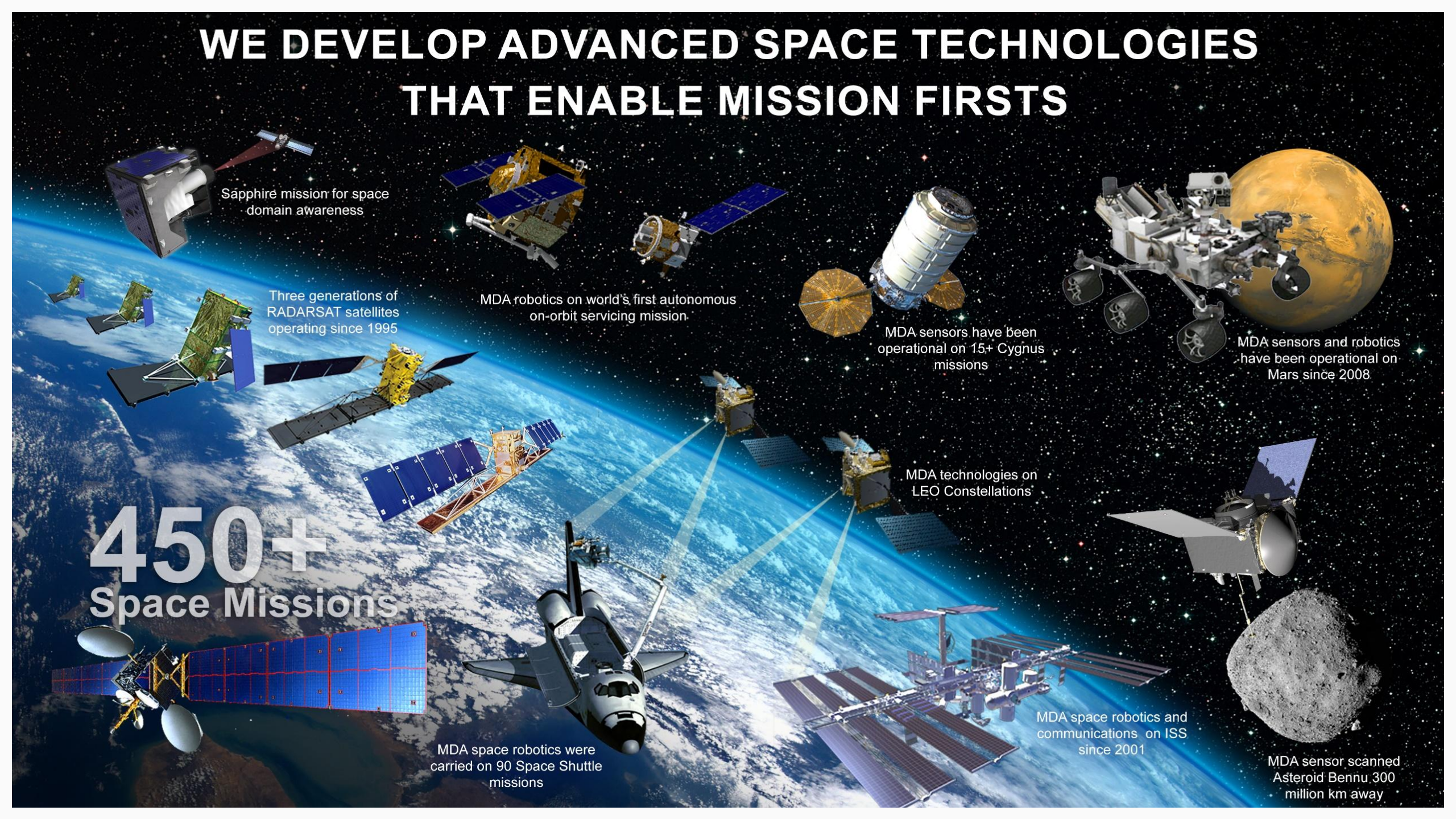
MDA technologies on
LEO Constellations

450+
Space Missions

MDA space robotics were
carried on 90 Space Shuttle
missions

MDA space robotics and
communications on ISS
since 2001

MDA sensor scanned
Asteroid Bennu 300
million km away



MDA Space is currently developing an advanced portfolio of robotics products and services

A range of products and services to meet a growing space industries' needs



Robotic Manipulators

Derived from peerless Canadarm technology, the MDA SKYMAKER™ suite of commercial robotics and services is on pace to enable a new commercial space economy like never before.



Surface Mobility

Covering a range of offerings from full mobility platforms to subsystems, MDA is developing GNC, locomotion, avionics, and payload interfaces to reliably get you from A to B and everywhere in between.



Sensors & Cameras

Lidar and vision system products support traversing even the harshest environments and terrains.



Space Operations

A new state of the art Mission Control Center is set to support government and commercial programs, giving you the mission support and training you need so you don't have to worry about the robotics part.



Purpose of the Presentation

- At the ESA Software Product Assurance Workshop 2023 (ESAC), we presented the problem description of qualifying Existing Software for Safety-Critical Ground Segments.
- Of the proposed possible solutions, the Product Service History (PSH) solution was selected for further investigation.
- The results of the first phase of this PSH activity are discussed in today's presentation.



Reuse of Existing Software

- Definitions

Existing Software (Gateway External Robotics System Product Assurance Requirements for Reuse of Existing Software, 4020749 Rev. B)

Any software developed outside of the Gateway project development as is or with adaptation. It includes software from previous developments provided by the supplier, software from previous developments provided by the customer, COTS, GOTS and MOTS software, freeware and open source software.

Software Reuse (NASA Software Engineering Requirements, NPR 7150.2D)


A software product developed for one use but having other uses or one developed specifically to be usable on multiple projects or in multiple roles on one project. Examples include, but are not limited to, COTS products, acquirer-furnished software products, software products in reuse libraries, and pre-existing developer software products.



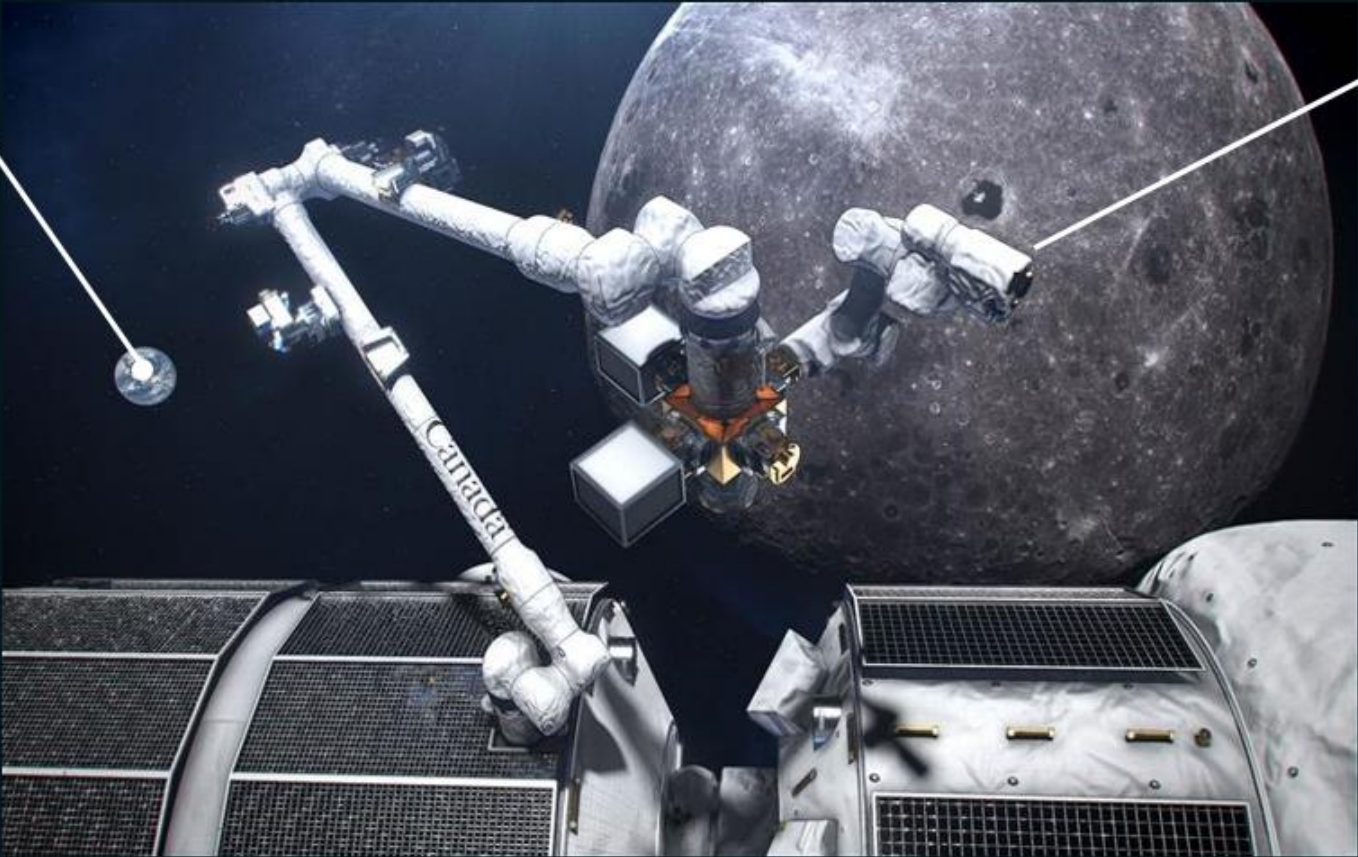
Safety-Critical Ground Systems (1/3)



Robotics Ground Segment

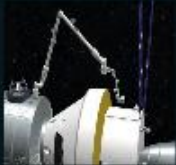


AI Enabled Autonomous Operations

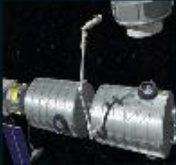


[Image Credit: Canadian Space Agency]


Robotics Flight Segment




Inspection & Repair




Vehicle Capture




Self-Maintenance



Assembly



EVA Support



Science



Safety-Critical Ground Systems (2/3)





Safety-Critical Ground Systems (3/3)

- Examples of Safety-Critical Functions:
 - Action overrides
 - Protection against cyber attacks
 - Display of time-critical telemetry to operators
 - Transmission of critical telecommands and data to the Flight Segment
 - Functions supporting Flight Segment autonomy:
 - Mission planning (task & path planning)
 - Collision Avoidance model verification and certification
 - Hardware-in-the-Loop (HIL) Simulations



Software Reuse in Ground Systems (1/2)

- Large ground operation systems rely heavily on software reuse for fundamental functions, including:
 - Operating systems like Windows or Linux
 - Development frameworks like .NET and Java
 - Visualization frameworks like Angular and React
 - Audiovisual frameworks like FFMPEG and VideoLAN
 - Browsers like Chrome or Firefox
 - Virtualization platforms like Docker and Kubernetes
 - Databases like MySQL and MongoDB
 -





Software Reuse in Ground Systems (2/2)

- Problem description
 - The integrated software product, including all reused software components, supports safety-critical functions.
 - While the custom-developed parts of the system may be qualified to the appropriate software class, the integrated system remains qualified at the lowest class of any of its components.
 - How can we (delta-)qualify software products like Windows, .NET, Docker, ...?
 - Most of the reused software products were never intended for safety-critical applications.
 - Can we actually rely on these software products to support safety-critical functions?
 - At the same time, the in-house development of these software products is not feasible and may arguably also lead to less reliable outcomes.



Product Service History (PSH)

- PSH is defined as “the utilization of the information about previous in-service experience of the component that is relevant to the new intended application and that can constitute evidence of product dependability.” (ECSS-Q-HB-80-01, Rev. A)
- Can be achieved by:
 - a) collecting existing information about previous in-service experience;
 - b) generating and collecting in-service experience; or
 - c) a combination of (a) and (b).
- In-service experience for PSH is only acceptable if the context of previous use is known, well-defined and similar to the context of usage of the new intended application.
- In case of differences between the previous context(s) of usage, the gaps shall be documented and identified, so that delta-qualification activities can be established, if needed.

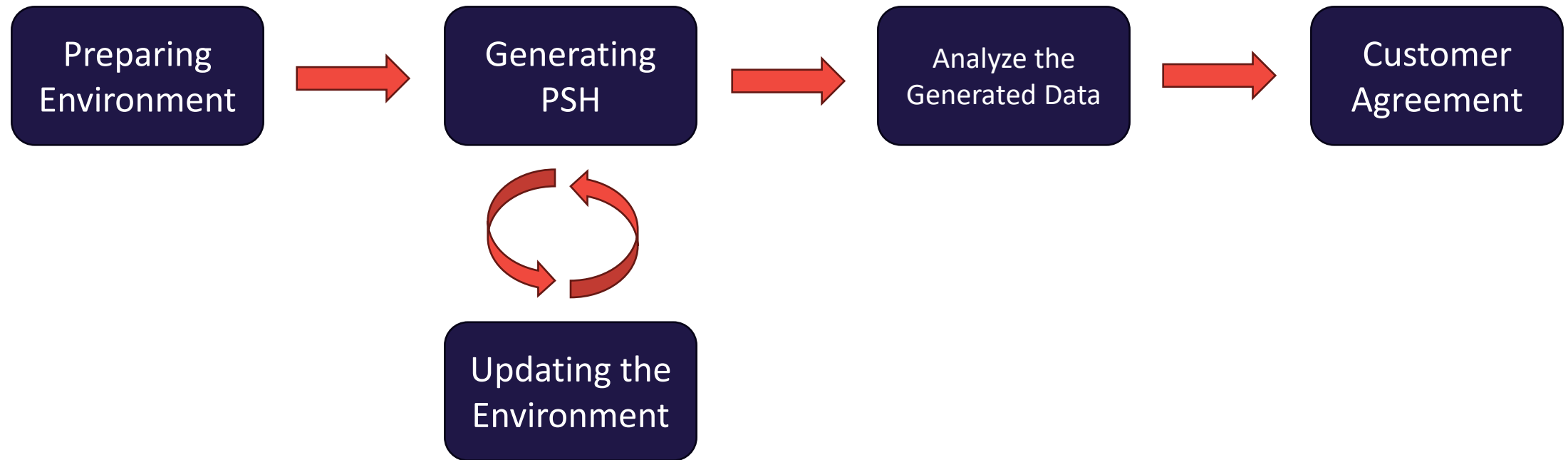


PSH for the Canadarm3 Ground Segment

- **Generation** of service history in parallel to development using analogue environments
- Steps:
 1. Investigate and collect any existing service history of the individual reused packages used in the Ground System. In case product versions are different, the delta and its potential impact on safety and dependability shall be documented.
 2. Configure an analog environment which uses all these packages, in a configuration that mimics the Ground System as closely as possible. Any deltas and their potential impact on safety and dependability shall be documented.
 3. Generate service history for the product built in (2), by deploying it in a context similar to the operational context of the GERS GS.
 4. Collect and documenting adequate service history, in accordance with the guidelines of ECSS-Q-HB-80-01, Rev. A, section 7.6 and Annex B, and DOT/FAA/AR-01/125. The service history shall be collected over an adequate length of time, per ESSB-HB-Q-002.
- Approach approved by the Gateway Software Control Panel.

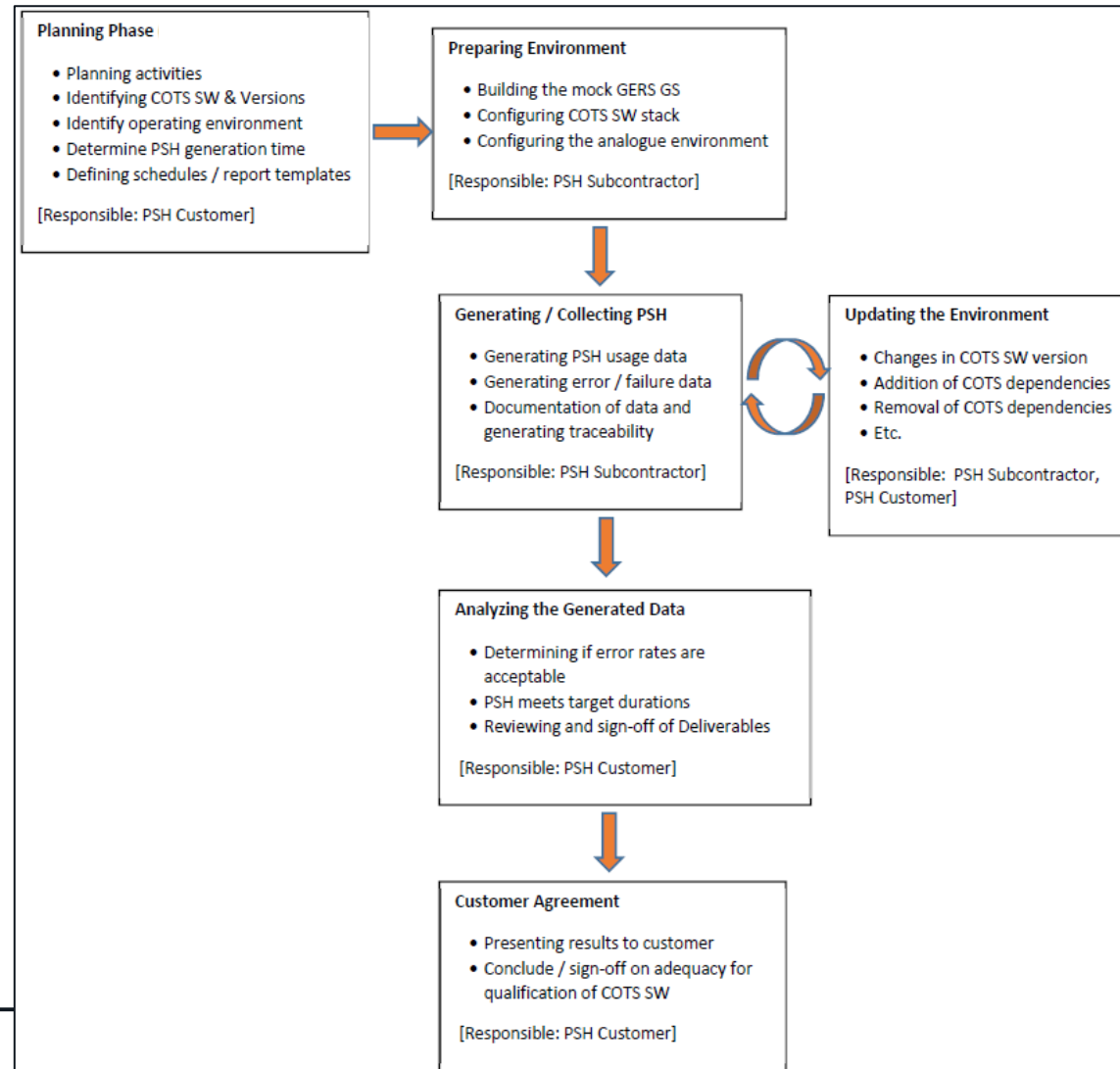


PSH Generation Lifecycle (1/2)





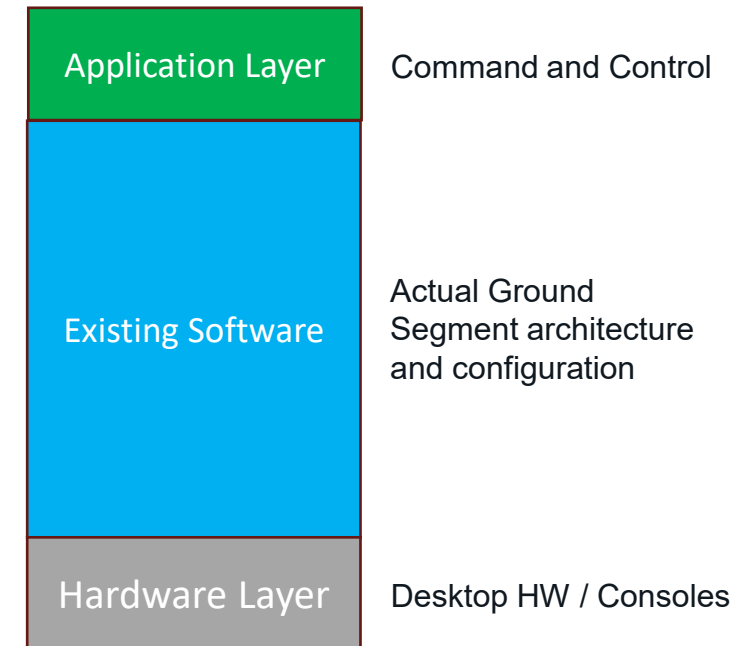
PSH Generation Lifecycle (2/2)





Generating PSH: Ensuring Platform Similarity

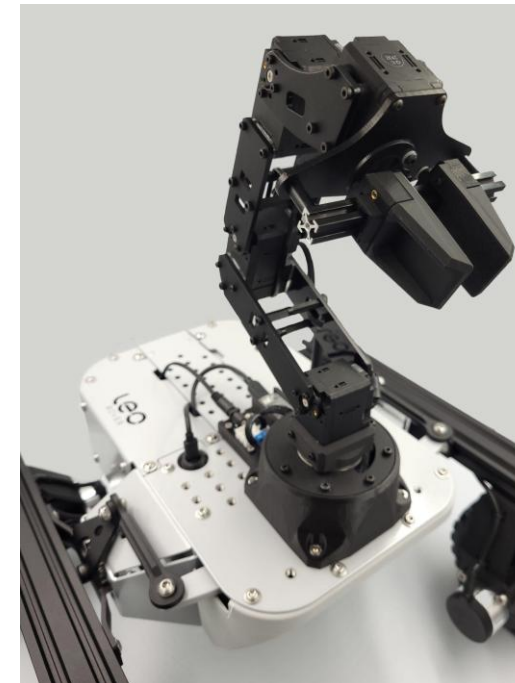
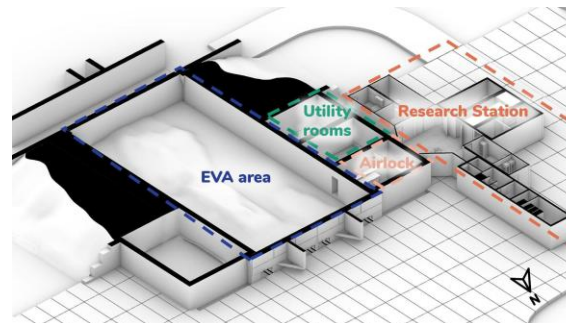
- PSH Configuration:
 1. A Hardware Layer
 - Limited to the existing Analog Facility hardware with modifications, where necessary, to run the Existing SW stack.
 - Representativeness of hardware must be analyzed and understood.
 2. The stack of Existing software
 - The Existing SW stack is configured in the same manner (using the exact same versions) or as closely as possible to its configuration on the actual Ground Segment.
 3. Application layer
 - Comprised of existing Analog Facility application SW, which is used as-is or modified minimally to work with the Existing SW stack, where necessary.
 - The application layer must be capable of executing typical, representative, robotic mission operations.





Generating PSH: Ensuring Operational Similarity (1/3)

- PSH generation executed in the LunAres analog facility operated by Space Is More (Piła, Poland)



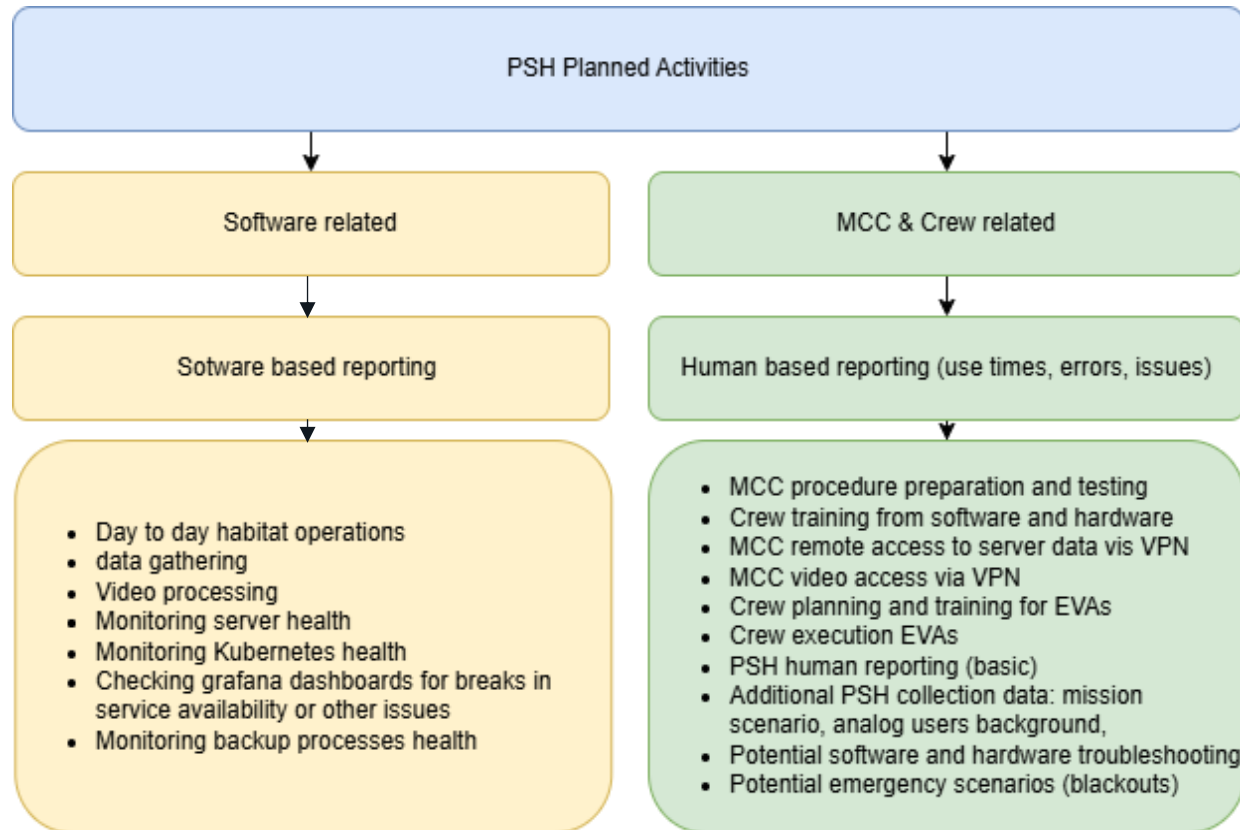


Generating PSH: Ensuring Operational Similarity (2/3)

- The SW integrated on the PSH Configuration was exercised to include the following types of representative operations:
 - Planning
 - Simulation
 - Robotics motion
 - Data analysis
 - Data storage
 - Verification support
 - Crew and Operator training
 - Video Recording and Playback including live streaming



Generating PSH: Ensuring Operational Similarity (3/3)





Generating PSH: Error Detection, Recording and Reporting

- Reporting includes (per ECSS-Q-HB-80-01 and DOT/FAA/AR-01/125) :
 - General SW information
 - Software analysis
 - Comparison with current project software standard
 - Hardware environment analysis
 - Operating environment analysis
 - Length of service period
 - Anomaly definition and anomaly rate
 - PSH data feeding
 - Configuration management process
 - PSH raw data collection
 - Anomalies report
 - Anomalies estimation
 - Stability and maturity of the product
 - Version configuration report

Table 8: Anomaly rate estimation

	During Overall validation	Expected value on a 3 months duration	Expected value on a 6 months duration
Total number of modifications			
Number of problem reports			
Major			
Minor			
Number of evolutions			
Global anomaly rate			
Anomaly rate of problem reports (Major)			
Anomaly rate of problem reports (minor)			
Anomaly rate of evolutions			
Ratio of COTS SW modules impacted by modification			
Ratio of COTS line of code impacted by modifications			
Number of COTS upgrade performed			

Table 9: Anomaly rate versus time

	From... to...	From... to...	From... to...
Total number of modifications			
Number of problem reports			
Major			
Minor			
Number of evolutions			
Global anomaly rate			
Anomaly rate of problem reports (Major)			
Anomaly rate of problem reports (minor)			
Anomaly rate of evolutions			
Ratio of COTS SW modules impacted by modification			
Ratio of COTS line of code impacted by modifications			
Number of COTS upgrade performed			



Results

Table 6-2: PSH SW Failures By Existing SW Product

SW Item Anomaly / Failure Detected	# Instances
Windows OS [Windows 10 Enterprise 2H22]	50
Chromium (Browser Engine)	1
Edge (Browser, Chromium-Based) [124.0.2478.80 (64 bit)]	1
Chrome (Browser, Chromium-Based) [124.0.6367.156 (64 bit)]	1
Docker [23.0.6, build ef23bc]	1
Couchbase Server [server-7.6.2]	3

SW Item Anomaly / Failure Detected	Failure / Anomaly	Failure Notes / Log Description	Space is More Conclusion / Assessment	# of Instances
Windows OS [Windows 10 Enterprise 2H22]	Application Error	Detected Anomaly: Application Crashes and System Logging Failures. Application Error (Event ID: 1000) at 04:07:40, indicating a crash in dptf_helper.exe.	Potential impact on client usage: unexpected service crash may lead to instability or data loss under load.	1
	Application Error	Detected Anomaly: iVMS-4200 Access Controller Failure. Application Error (Event ID: 1000) indicating a crash of iVMS-4200.AccessController.S.exe. May be linked to software instability, memory corruption, or system resource exhaustion. Classification: Software Anomaly - iVMS-4200 Application Crash.	Potential impact on client usage: access-controller crash will interrupt video streams and monitoring features for end users.	2
	Detected Anomaly: Persistent Runtime Corruption	Detected Anomaly: Persistent Runtime Corruption Multiple occurrences of AppModel-Runtime errors (Event ID: 80) indicating corrupted Microsoft package family runtime information. Errors occurred between 09:50:10 and 09:50:27, and again at 09:56:42, showing a repeated failure pattern.	Potential impact on client usage: corrupted runtime may cause application crashes or loss of functionality under load.	1

Note: These are the failures that are considered to have a potential impact on GS operations per the Subcontractor. There are other failures encountered during PSH generation, which were analyzed and deemed to have "no impact" on GS operations.



Next Steps

- Review the failure rates per the tables above.
- Conclude on the failures / anomalies in the COTS SW products identified encountered and identify the risk of these errors with respect to Canadarm3 GS operation.
- Propose alternative actions or measures to mitigate these risks, including but not limited to the following:
 - Propose a different version of the COTS SW product
 - Propose an alternative to the COTS SW product
 - Propose application software changes to mitigate the errors
 - Justify use as-is, including risk assessment
- Get customer approval of the reuse qualification achieved through PSH generation.



So What?

- We have demonstrated the using PSH generation for the qualification of Existing Software in safety-critical ground segments is feasible.
- While it involves a lot of advance planning and logistics, as well as discipline during lengthy development lifecycles, it can lead to compliance with existing safety and PA standards.
- Further evolving and refining this approach can lead to the cost-effective implementation of safe, reliable, versatile and future-proof ground segments.



Q&A



THANK YOU

CONTACT INFO:

SAFETY & MISSION ASSURANCE

EMMANUEL.LESSER@MDA.SPACE

TIM.ARULSUTHAN@MDA.SPACE

