



DO-178C Services + AI

Ed Liversidge

Director, CTO

Sukhvir Purewal

Sales Director



We fix, develop and test embedded software for the aerospace & defence industry specialising in DO-178C safety-critical testing

- Formed in 1999
- Based near Horsham, UK
- SME of 6 people (and growing)
- We work with:
 - Leonardo, Wind River, Green Hills Software, IBM, Curtiss Wright, Abaco Systems, U-TACS
 - Babcock International, GE plus more...



OUR AEROSPACE PROJECTS 1

DO-178C VxWorks 7 Level A certification for Avionics CPU

- Certified VxWorks 7 for PPC T2081 flight computer.
- Ported a custom run time patching test harness from ARM to PPC.
- Ran > 10K tests.
- 100% code coverage, using LDRA.
- Fixed test failures using Polarion to manage problem reports.
- Liaised with other teams worldwide.





OUR AEROSPACE PROJECTS 2

DO-178C Bare Metal Level B certification for Avionics System

- Testing Boot Loader and Power on Built-In Test.
- Involved creating compile time test harness from scratch.
- 100% code coverage, using VectorCAST.
- 100% requirements-based test coverage, using Lauterbach debugger.
- PBIT tested PCI, Ethernet, 1553, Arinc429 NOR Flash etc.
- Manage problem reports using Jira.
- Liaised with other teams in UK and USA.



PROBLEM 1

In the **\$90 Billion** aerospace **avionics**¹ market, the **certification**² of safety-critical software is:



Expensive (£2M+)



High project & life risk



Slow (2+ years)



Labour intensive (10+ people)

1. Aircraft + electronics

2. Certification demonstrates the reliability and safety of software systems, minimising loss of life.

PROBLEM 2

Certification **suffers** from:



Lack of coherent tools



Ever increasing complexity



Intense time pressures



Tedious, labour intensive, repetitive work



Staff attrition & lack of experience



PROBLEM 3 - TOOLS

DO-178C can use a mix of disconnected process management tools e.g:

- Requirements managements tool
- Word for documents (PSAC, SDP, SVCP etc)
- Excel for reviews and traceability
- Perforce and/or Git for version control
- Jira or Polarion for Problem Reports
- Random code editors

Disadvantages:

- Stand alone requirements tool can be difficult to integrate with other tools
- Change control (specifically diffs) difficult with Word and Excel
- No links possible between these tools and meta data
- Emails get lost, reviews are slow and tedious
- Engineers not leveraging powerful editors

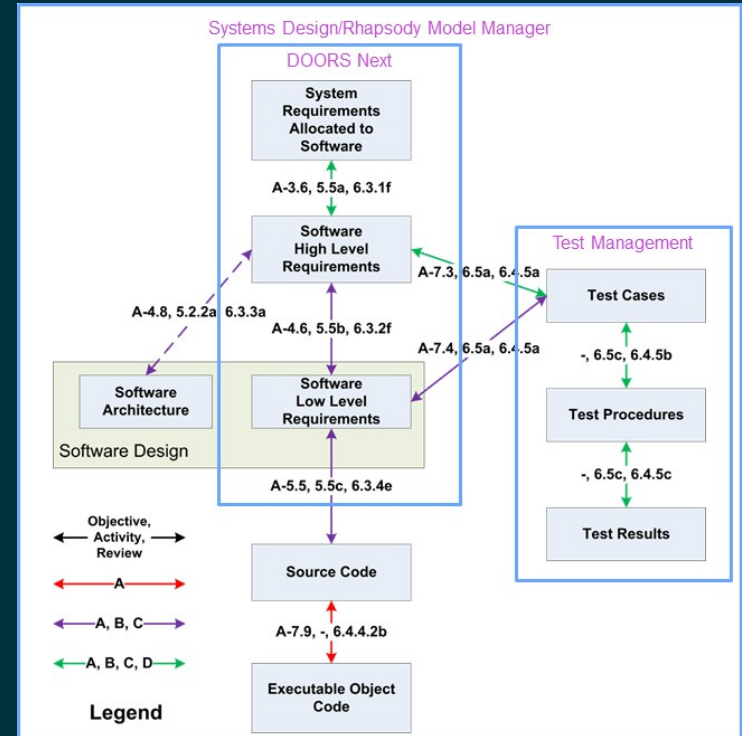
😊 SOLUTIONS 1

We offer two solutions:

✓ Enterprise : IBM's Enterprise Lifecycle Management

A framework of inter-connected tools, tuned for DO-178C:

- DOORS (requirements)
- Rhapsody (Model Based System Engineering)
- Workflow (project planning and defect tracking)
- Test management
- **Contact us** for an ELM Demo



SOLUTIONS 2

We offer two solutions:

- ✓ **Small-Medium** : **Spacemacs** + **Git** + **Jenkins** + **AI**
 - Free open source tools utilising Plain Text.
 - **Spacemacs** : Powerful configurable text editor with > 40 years of development.
 - **Git** : Industry standard source code control and tracking.
 - **Jenkins** : Automated test management.
 - **AI** : Powerful Large Language Models trained on DO-178C, industry best practices and customer data and procedures.

DO-178C IN PLAIN TEXT 1

Proposal:

All DO-178C meta data, including all documents, traceability, reviews, checklists, source code, test code etc, stored as Plain Text and managed like source code, with deep linking.

Advantages of Plain Text:

- ✓ Timeless
- ✓ Always readable
- ✓ Searchable (grep, ag, rg, fzf)
- ✓ Parseable (Python¹ and AI / RAG²)

1. Parsing SVCP for expected results vs actual results for automated tests

2. Retrieval Augmented Generation – deep AI knowledge base

Advantages of Plain Text + Git:

✓ Track every change, update, review etc

✓ Distributed

✓ Powerful Git integration with Spacemacs

Disabling MMU on ARM

✨ AI Generated!

```
* OPEN HSS-IMX8-BL-IMPL-LLR001.01.04 _start shall disable the MMU
Here is the assembly snippet that clears the MMU enabled bit (bit 0) of SCTLR:

```assembly
MRC p15, 0, r0, c1, c0, 0 ; Read SCTLR into r0
BIC r0, r0, #1 ; Clear the MMU enabled bit (0-th bit)
MCR p15, 0, r0, c1, c0, 0 ; Write back the modified value to SCTLR
```

In this example, `MRC` (Move Register to Coprocessor) is used to read the SCTLR into the general-purpose register `r0`. Then, the `BIC` (Bit Clear) instruction clears the least significant bit (bit 0). Finally, the `MCR` (Move Coprocessor to Register) instruction writes back the modified value to SCTLR.

Ensure that this code is executed in a privileged mode, such as supervisor or system mode, because modifying SCTLR requires privileged access.

* OPEN HSS-IMX8-BL-IMPL-LLR001.01.05 _start shall disable the instruction cache
:PROPERTIES:...

implements : HSS-IMX8-BL-HLR016 The Bootloader shall disable all caches
verified by : HSS-IMX8-BL-VC-LLR001.01.05

To disable the instruction cache in a Cortex-A53 processor using ARM assembler, perform the following steps:

1. Read the System Control Register (SCTLR) into a general-purpose register.
2. Clear the I-bit (bit 12) of the SCTLr to disable the instruction cache.
3. Write the updated value back to the SCTLr.
4. Invalidate the instruction cache to ensure coherence.

Example code to implement the above steps:

```assembly
MRS R0, SCTLR // Read SCTLr into R0
BIC R0, R0, #(1 << 12) // Clear the I-bit to disable instruction cache
MSR SCTLR, R0 // Write back to SCTLr

// Invalidate entire instruction cache
MOV R0, #0
MCR p15, 0, R0, c7, c5, 0 // Invalidate ICaches to Point of Unification
```

>Head: master added org-roam instructions
Rebase: origin/master added org-roam instructions
Push: origin/master added org-roam instructions

>Untracked files (11)
>Unstaged changes (2)
  modified 03-Design/HSS-IMX8-BL-FILE-LLR001.org
  @@ -3,7 +3,7 @@
  :END:
  #+TITLE: HSS-IMX8-BL-FILE-LLR001
  #+CREATED: 2023-11-16
  -#+TODO: TODO TODO-AI TODO-REVIEW TODO-AI-REVIEW OPEN | WAIT
  +#+TODO: TODO TODO-AI TODO-REQ-REV TODO-DES-REV TODO-AI-REQ-
  #+STARTUP: overview

[[implements : [[id:2f71749b-13a2-449f-a722-4aa1a04fdb2c]][HSS
  @@ -71,7 +71,7 @@ CLOSED: [2023-11-19 Sun 12:07]
  >@@ -114,7 +114,8 @@ aligned, it calculates the required offs
  >@@ -126,7 +127,8 @@ requirement HSS-IMX8-BL-LLR003 necessita
  >modified 03-Design/HSS-IMX8-BL-FILE-LLR013.org

>Recent commits
bb12410 * [origin/master] added org-roam instructions
f0adb75 * Added TODO-REVIEW to TODO states
90a8f7c * Updated assignments
f362de1 * renode resc script
4f0953b * Updates to Branch Symbol / bootloader code
7136fe7 * Assigning FILE-LLR001 to jjohn
131294f * Project reassignment to Jerin and Dan
7f872f3 * start.S code updates
434dd20 * Updated elisp code to be directory independent
a6998a8 * Added write-vc
```

Advantages of Plain Text + Git + Spacemacs:

✓ Org Mode

- Powerful markup language
- Linking between all meta data
- Export to html, pdf
- Task and progress tracking with timestamps
- Agenda view
- Design diagrams (dynamically regenerated)

```
#+TITLE: HSS-IMX8-BL-FILE-LLR001
#+CREATED: 2023-11-16
#+TODO: TODO TODO-AI TODO-REQ-REV TODO-DES-REV TODO-AI-REQ-REV TODO-AI-DES-REV OPEN |
#+STARTUP: overview

implements : HSS-IMX8-BL-HLR009 The Bootloader shall initially execute ARM assembler s
  o HSS-IMX8-BL-FILE-LLR001 There shall be a file called start.s
  o HSS-IMX8-BL-FUNC-LLR001.01 start.s shall implement the entry point function in the
    :PROPERTIES:
    :verified-by: HSS-IMX8-BL-VC-LLR001.01
    :
  * DONE Sequence Diagram
  CLOSED: [2023-11-19 Sun 12:07]
  -#+BEGIN_SRC dot :cmd dot :results file link :file ./images/start_s.png :wrap "0

#+RESULTS:
#+begin_org
#+CAPTION: Function _start
#+LABEL: fig:_start
#+ATTR_HTML: :width 500
```

```
graph TD
  reset[/reset/] --> LLR001_01_01[LLR001.01.01  
Check 4K Alignment]
  reset --> LLR001_01_06[LLR001.01.06  
Disable Data Cache]
```




MACHINE LEARNING 2 – “VIRTUAL DER”

Question: “What are baselines and why are they important. Give references.”

Answer: Baselines are specific versions of configuration items that have been formally reviewed and agreed upon at a certain point in time and that serve as the basis for further development. They are important because they provide a stable foundation for ongoing development and maintenance activities, ensuring that changes are made in a controlled and systematic manner.

Baselines are important for several reasons:

1. **Control:** They help in controlling the evolution of software products by providing reference points that changes can be compared against (7.2.2a).
2. **Integrity:** Establishing baselines in controlled libraries ensures the integrity of the software, as it protects the baseline from unauthorized changes (7.2.2c).
3. **Traceability:** Baselines provide traceability, allowing each configuration item to be traced back to its origins, which is crucial for understanding the evolution of the software and for maintaining it effectively (7.2.2e, f, g).
4. **Change Management:** They facilitate effective change control by ensuring that any changes to configuration items are recorded, approved, and tracked (7.2.4b, c).
5. **Certification:** For software that is used in critical applications, such as airborne systems, baselines are essential for certification purposes. They must be established for configuration items used for certification credit, and changes must be managed to maintain certification (7.2.2a, 10.1).
6. **Upgrading:** When upgrading software, especially in the context of certification, it is important to establish a new baseline that takes into account the previous development baseline and the objectives of the new application (12.1.4a).

In summary, baselines are a fundamental part of software configuration management and are critical for maintaining the stability, integrity, and traceability of software products throughout their life cycle. They are particularly important in environments where software must be certified or where it is critical to system safety and functionality.



MACHINE LEARNING 3 – REQUIREMENTS ANALYSIS

Inadequate Requirements → Expensive Rework!

Solution, use AI to:

- ✓ ensure correct formatting, positive, atomic, active voice etc.
- ✓ ensure correct HLR/LLR detail
- ✓ ensure compliance with the SW Requirements Specification

```
REVIEWED-AI HSS-IMX8-BL-IMPL-LLR001.01.04 startup.s shall disable the MMU
FAIL Review comments for HSS-IMX8-BL-IMPL-LLR001.01.04 startup.s shall disable the MMU
CLOSED: [2024-03-06 Wed 10:56]

0. Correct Structure: PASS
1. Contains shall: PASS
2. Non Negative: PASS
3. Active Voice: PASS
4. Does not use 'shall be able to': PASS
5. Grammar & Spelling: PASS
6. Atomic: PASS
7. LLR Check: FAIL

The discussion revolves around the analysis of a specific requirement related to a software startup script, designated as "startup.s," which mandates the disabling of the MMU (Memory Management Unit). Initially, it's noted that the structure of this requirement adheres to a format where 'startup.s' serves as the subject, 'shall disable' acts as the action to be taken, and 'the MMU' is the object of this action. This formulation does not explicitly include conditions or constraints.

The placement of the word 'shall' immediately after the subject, as observed in the requirement "HSS-IMX8-BL-IMPL-LLR001.01.04 startup.s shall disable the MMU," is recognized as correct according to specified rules. The requirement is identified as a positive statement, indicating a directive for an action rather than prohibiting one, and is phrased in an active voice. It is also observed to utilize clear, unambiguous language, avoiding vague terms like 'shall be able to,' 'timely,' 'fast,' 'sufficient,' etc., and is grammatically sound without any spelling mistakes. Additionally, the requirement is noted to be atomic, meaning it specifies a single, clear action.

However, while the requirement effectively identifies the necessary action (disabling the MMU in "startup.s"), it is pointed out to lack detailed instructions or the specificity needed for implementation, particularly for engineers who may not be familiar with the processor architecture or the specific procedures to disable the MMU. This means it directs what needs to be done, without clarifying how it should be achieved, potentially leaving room for interpretation or necessitating further clarification for proper execution.
```


MACHINE LEARNING 4 -SQA CHECKLISTS

DO-178C has many checklists, each containing multiple questions. Our AI can perform reviews on demand:

 gives confidence, reduces risk, saves time

```
#+TITLE: Plan for Software Aspects of Certification (PSAC)
#+AUTHOR: [Ed Liversidge]
#+Document ID: [123-456789-PSAC-000]

◎ 1. Introduction...
◎ 2. System Overview...
◎ 3. Software Overview...
◎ 4. Certification Considerations
  ◎ 4.1 Certification Basis & Means of Compliance
    The software level (A-E) for the software component will be determined in accordance with DO-178C standards, impacting the rigor of development and verification efforts. Higher software levels require more stringent efforts to ensure compliance with airworthiness requirements, as software at Level A, if it were to behave anomalously, could contribute to a catastrophic failure condition for the aircraft.

    In our case, the system must be certified to DO-178C Level B (DAL B).

    Justification from the DO-178C standard for Level B is shown below:

    Level B: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a hazardous failure condition for the aircraft.
```

```
REVIEWED-AI #11 Is the software DAL level stated, along with justifications to support the choice of DAL level?
  ○ PASS Checklist 11 gpt-4-1106-preview AI review comments
    CLOSED: [2024-03-06 Wed 13:39]

    The provided information states that the Design Assurance Level (DAL) is Level B and that justifications for this choice are based on the system safety assessment process. The document asserts that the potential impact of software anomalous behavior on the system function could lead to a hazardous failure condition for the aircraft, which complies with the DO-178C standard for Level B certification. This standard is applicable for software that, if not performing correctly, would result in a failure causing or contributing to a hazardous condition.

    Given this information, the document shows compliance by defining the DAL as Level B and by providing a rationale that aligns with the requirements of the DO-178C standard regarding the consequences of software failure.
```



MACHINE LEARNING 5 – MULTI AGENT AI

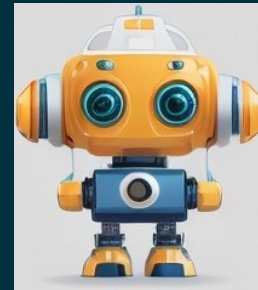
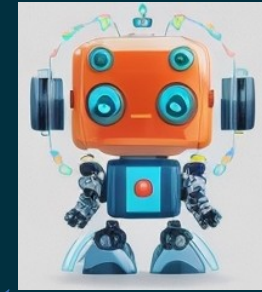
Utilise **Multiagent AIs**, trained on:

- Certification process
- Customer data and processes
- Technical manuals
- Customer project

They communicate to:

- Write and review docs
- Write and review code
- Run tests and verify results
- Execute cert checklists

PROJECT MANAGER



PROJECT EXPERT



TECH EXPERT



CERT EXPERT





THE BENEFITS TO YOU

We utilise our **always on** group of trained **AI experts** (on-site and off-site) to:

- ✓ Shorten timescales & reduce cost
- ✓ Reduce risk
- ✓ Improve efficiency
- ✓ Do the tedious work, on demand

? POTENTIAL ISSUES - ADDRESSED

Security

- We have on-site LLMs, prevents sending data to OpenAI
- Partnership with IBM's **WatsonX** AI for security and governance

Difficult to get consistency, LLMs can hallucinate

- We have an automated test suite to provide metrics and improvements
- Humans make mistakes as well!

Can use a lot of tokens

- still < \$10 per day

Cannot be qualified as a tool, but doesn't need to be:

- it does not make safety-critical decisions
- all output is, and always will be, subject to final human review

DO-178C SERVICES + AI

What Is The Problem We Solve?

The most stringent, difficult and rigorous level of development and testing in the DO-178C standard. It requires the highest levels of expertise, resource and experience to meet compliance, quality assurance and certification.

What Is The Solution We Provide?

We simplify DO-178C safety-critical development for clients by providing a third-party service that handles the stringent safety testing, extensive verification, traceability, fault tolerance, systems integration, documentation and compliance.

Why Use Us For DO-178C Projects?

Along with utilising an in-house cutting-edge AI software toolset, designed entirely around the needs of DO-178C, the service bridges a skills gap and brings to the table considerable expertise and experience from prior DO-178C projects. By leveraging our mastery in DO-178C development and testing, clients reduce cost, mitigate risk and deliver high-quality, certified systems.

Got A Project In Mind?

We fix, develop and test embedded software for the aerospace & defence industry specialising in DO-178C safety-critical testing. Speak to us. We have laser focus, unprecedented attention-to-detail and provide phenomenal value.



HARMONIC
SOFTWARE SYSTEMS

Schedule a Call To Skyrocket Your DO-178C Process

✨ [CLICK HERE TO SCHEDULE A DEMO](#) ✨

Phone : +44 (0)1403 784500

Email : sales@harmonicss.co.uk