SPAZIO IT

OPEN SOURCE & AVIONICS

Maurizio Martignano
Spazio IT – Soluzioni Informatiche s.a.s
Via Manzoni 40
46030 San Giorgio di Mantova, Mantova
http://www.spazioit.com

December 2014

Agenda

Agenda

- Who am I?
- Avionics
- Standards (DO-178(B)C)
- RTEMS
- SonarQube
- (Open source) licenses and intellectual property
- Qualification and certification
- Maintenance and support
- Economic Sustainability

Who am I?

Who am I?

- Name: Maurizio Martignano
- Experience in avionics:
  - 1986-1988: software for military planes radar system (FIAR)
  - 1991-1999: Spacecraft on-board software (European Space Agency),
  - 2002-2007: software systems used by the crew on-board the International Space Station(European Space Agency)
  - 2010-2012: work on a space qualified version RTEMS (Galileo Project)
  - 2012-2013: development of a SonarQube Ada Plugin for AIRBUS Helicopters
  - 2013-2014: development of a SonarQube C/C++ Plugin in support of IVV activities for on-board software (European Space Agency)

Who am I?

- Experience in Open Source
  - GCC-1750 (1995-2006)
    http://www.esa.int/TEC/Software_engineering_and_standardisation/TEC8VAUXBQE_0.html
  - OpenACS (2006-Now)
    http://www.openacs.org/
  -]project-open[(2009-Now)
    http://www.project-open.org/en/list_partners
    http://sourceforge.net/projects/project-open/files/project-open/V4.0/
  - Oracle ADF (2011-Now)
    http://www.spazioit.com/pages_it/sol_inf_it/oracle_adf_it/

Who am I?

- Subject related publications:
  - Martignano, Gaisler, Nettleton, “GNU Based Compilation Systems for Spacecraft Microprocessors”, Data Systems in Aerospace 1997, SEVILLA, Spain, May 1997.
  - Martignano, “GNU Based Compilation Systems for Spacecraft Microprocessors”, Preparing for the Future Vol. 7 No. 3, September 1997.
  - Blondin, Martignano, “GNU Based Compilation Systems for Space Applications”, Data Systems in Aerospace 1998, Athens, Greece.

Avionics

Avionics

- **Avionics** are the **electronic** systems used on **aircraft**, **artificial satellites**, and **spacecraft**.

- **Avionics** are the **electronic** systems used on **aircraft**, **artificial satellites**, and **spacecraft** as well as their (ground) **control center**.

Avionics Software

- Avionics software can be divided into three categories:
  - «Flight» Software (the software on-board the [space]craft)
  - «Ground» Software (the software in the control center)
  - Software Development Environment (the software used to develop flight and ground software)

Standards – DO-178(B)C

DO-178(B)C DALs

- **Software Considerations in Airborne Systems and Equipment Certification**
  - 5 Software Levels or Design Assurance Levels
  - **A: Catastrophic** – Failure may cause a crash. Error or loss of critical function required to safely fly and land aircraft.
  - **B: Hazardous** – Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the aircraft due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers. (Safety-significant)
  - **C: Major** – Failure is significant, but has a lesser impact than a Hazardous failure (for example, leads to passenger discomfort rather than injuries) or significantly increases crew workload (safety related)

DO-178(B)C DALs

- **D: Minor** – Failure is noticeable, but has a lesser impact than a Major failure (for example, causing passenger inconvenience or a routine flight plan change)
- **E: No Effect** – Failure has no impact on safety, aircraft operation, or crew workload.

DO-178(B)C DALs (the theory)

- Standards do require different activities, different types of testing, qualification, validation and verification, that must be executed on the software based on its criticality.

- But how about open source software? **Open source software can be used (A,B,C) if**:
  - **the source code is available** (ok - ☺)
  - **all testing, qualification, validation and verification activities that the standards require have to be performed also for the open source software** (very though, ☹ ☹ ☹)

DO-178(B)C DALs (in practice)

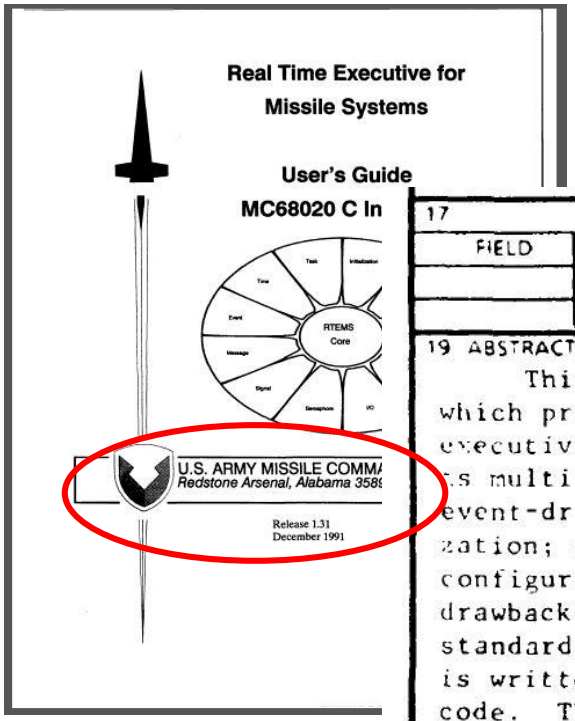
- Issues:
  - (Open source) licenses and intellectual property
  - Qualification and verification
  - Maintenance and support
  - Economic sustainability

- Possible Solutions
  - Literature (google)
  - Example: RTEMS
  - Example: SonarQube

RTEMS

Real
Time
Executive
for
~~Missile~~, ~~Military~~,
Multiprocessors
Systems

RTEMS – the beginning



17 COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	RTEMS, real-time, executive, heterogeneous, homogeneous, multiprocessing, 68020, microprocessor, C language, runtime, (Continued on page ii)

19 ABSTRACT (*Continue on reverse if necessary and identify by block number*)

This document is a detailed design manual for a real-time multiprocessor executive which provides a high performance environment for embedded military applications. This executive, known as RTEMS (Real-Time Executive for Missile Systems), includes such features as multitasking capabilities; homogeneous and heterogeneous multiprocessor systems; time event-driven, priority-based, preemptive scheduling; intertask communication and synchronization; responsive interrupt management; dynamic memory allocation; and a high level of user configurability. RTEMS was originally developed in an effort to eliminate many of the major drawbacks of the Ada programming language. RTEMS is based on the RTEID (now ORKID) proposed standard. The code is Government owned, so no licensing fees are necessary. The executive is written using the 'C' programming language with a small amount of assembly language code. The code was developed as a linkable and/or ROMable library with the Ada programming language. Initially RTEMS was developed for the Motorola 68000 family of processors. It (Continued on page ii)

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL	22b TELEPHONE (Include Area Code) 22c OFFICE SYMBOL

RTEMS - nowadays

- http://www.rtems.com
- The **Real-Time Executive for Multiprocessor Systems** or **RTEMS** is a open source fully featured Real Time Operating System or RTOS that supports a variety of open standard application programming interfaces (API) and interface standards such as POSIX and BSD sockets. It is used in space flight, medical, networking and many more embedded devices across a wide range of processor architectures including ARM, PowerPC, Intel, Blackfin, MIPS, Microblaze and more. […] We strive to provide regular, high quality releases, which we want to work well on a wide range of embedded targets using cross development from a variety of hosts including GNU/Linux, Mingw, MS-Windows, FreeBSD, Cygwin, and Solaris.
- Copyright (C) OAR Corporation
  7047 Old Madison Pike, Suite 320; Huntsville, AL 35806, USA

RTEMS - license

- (relaxed) GPL

- **As a special exception, including RTEMS header files in a file, instantiating RTEMS generics or templates, or linking other files with RTEMS objects to produce an executable application, does not by itself cause the resulting executable application to be covered by the GNU General Public License.** This exception does not however invalidate any other reasons why the executable file might be covered by the GNU Public License.

RTEMS – ESA / EDISOFT

- http://rtemscentre.edisoft.pt/
- Aiming at:
  - increasing in Europe the know-how on RTEMS
  - using a stable version of RTEMS (4.8.0)
  - removing from it everything which is not needed in a typical space mission (satellite on-board platform)
  - qualifying and certifying the software at DO-178C DAL B level
  - making the software available
  - guaranteeing its support

RTEMS – Spazio IT - IVV

- Spazio IT, in the context of RTEMS Centre, performed an IVV (Independent Validation and Verification) on the software made available by EDISOFT, and namely:
  - provided assistance while removing the «unnecessary» pieces of code
  - verified concretely, from a software engineering perspective, the quality of the produced code
  - measured in details the obtained results

RTEMS – Spazio IT - IVV

RTEMS [TRL6]

Subset of the baseline RTEMS 4.8.0

Quality level:

- Galileo Software Standards (Engineering, Configuration Management, Dependability, Product Assurance, Software Reuse and ISVV) Development Assurance Level – B
- SPEC/SPEC1 evaluation by TUV Rheinland InterTraffic GmbH
- ISVV by CAPTEC - Computer Applied Techniques (http://www.captec.ie/)
- ISVV by SPATIOIT - Soluzioni Informatiche s.a.s (http://www.spazioit.com/)
  - Statement Coverage 100% for C and Assembly Code for LEON 2 and LEON 3 Boards and ERC32, LEON 2 and LEON3 simulators
  - Decision Coverage 100% for C and Assembly Code for LEON 2 and LEON 3 Boards and ERC32, LEON 2 and LEON3 simulators

Users

- smallGEO
- Galileo
- Sentinel2
- IXV
- IMA for Space
- EarthCare

http://rtemscentre.edisoft.pt

Terraillon | Savoir Implementation SW | ADCSS | 25/10/11 | Pag. 17

ESA UNCLASSIFIED – For Official Use

RTEMS – Spazio IT - IVV

Tailored RTEMS

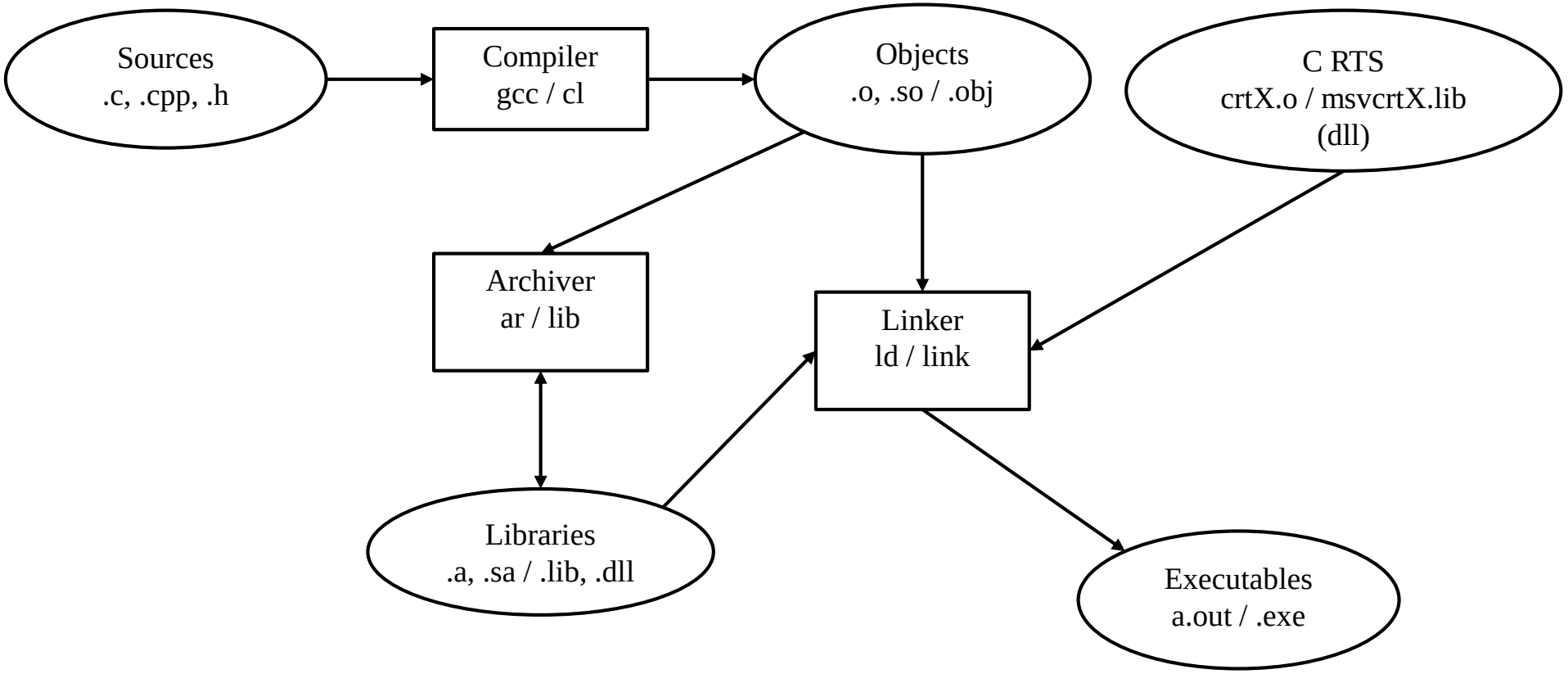
```
Running samples/base_sp.srec...


section .sec1 at 0x02000000 (33236 bytes)


*** SAMPLE SINGLE PROCESSOR APPLICATION ***

Creating and starting an application task

Application task was invoked with argument (0) and
has id of 0xA000002

*** END OF SAMPLE SINGLE PROCESSOR APPLICATION ***




Running samples/hello.srec...


section .sec1 at 0x02000000 (26244 bytes)


*** HELLO WORLD TEST ***

Hello World
```

RTEMS 4.8.0

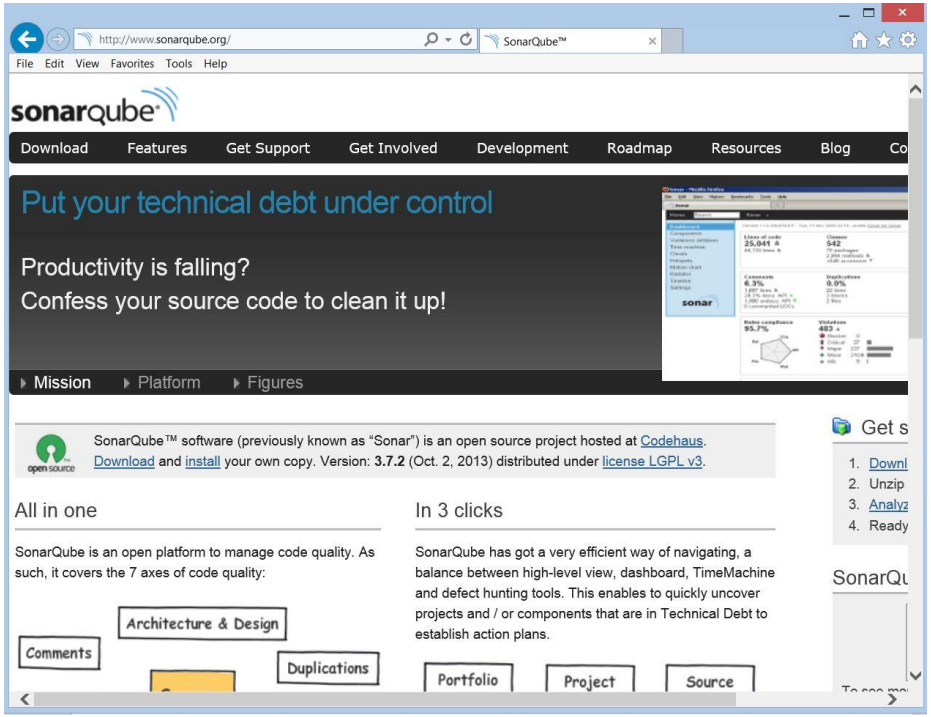
```
Running samples/base_sp.srec...


section .sec1 at 0x02000000 (98660 bytes)


*** SAMPLE SINGLE PROCESSOR APPLICATION ***

Creating and starting an application task

Application task was invoked with argument (0) and
has id of 0xa010002

*** END OF SAMPLE SINGLE PROCESSOR APPLICATION ***




Running samples/hello.srec...


section .sec1 at 0x02000000 (74068 bytes)


*** HELLO WORLD TEST ***

Hello World
```

Spazio IT – RTEMS - Demo

Spazio IT – RTEMS - Demo



SonarQube



SPAZIO IT

SonarQube – AIRBUS Helicopters

- In fall 2012 AIRBUS Helicopters (http://www.airbushelicopters.com) was looking for a code quality platform for the maintenance of the Tiger and NH90 software.
- Initially they only had available two solutions:
  - a solution proposed by the actual leader in code quality platform : very powerful solution but closed (not open source) and rather expensive
  - a solution proposed by the Munich Technical University: open source solution but rather limited in terms of functions/capabilities

SonarQube – AIRBUS Helicopters

- Spazio IT proposed to AIRBUS Helicopters a solution based on SonarQube (http://www.sonarqube.org/).
- Spazio IT solution got selected, instead of one of the original solutions.

SonarQube – Spazio IT

- NH90 and Tiger software is written in Ada (83).
- There was already an Ada Plugin for SonarQube but it did not work properly and eventually was removed by SonarSouce from the lists of available plugins.
- Spazio IT took the original plugin, made it work and added new functions to it.
- The plugin supports both static and dinamic analysis, is able to detect code duplications and interfaces with GNAT (AdaCore: http://www.adacore.com) compiler, Atego Apex (http://www.atego.com) compiler and with Understand (http://www.scitools.com).
- http://www.spazioit.com/pages_en/sol_inf_en/code_quality_en/ describes Spazio IT activities on code quality.

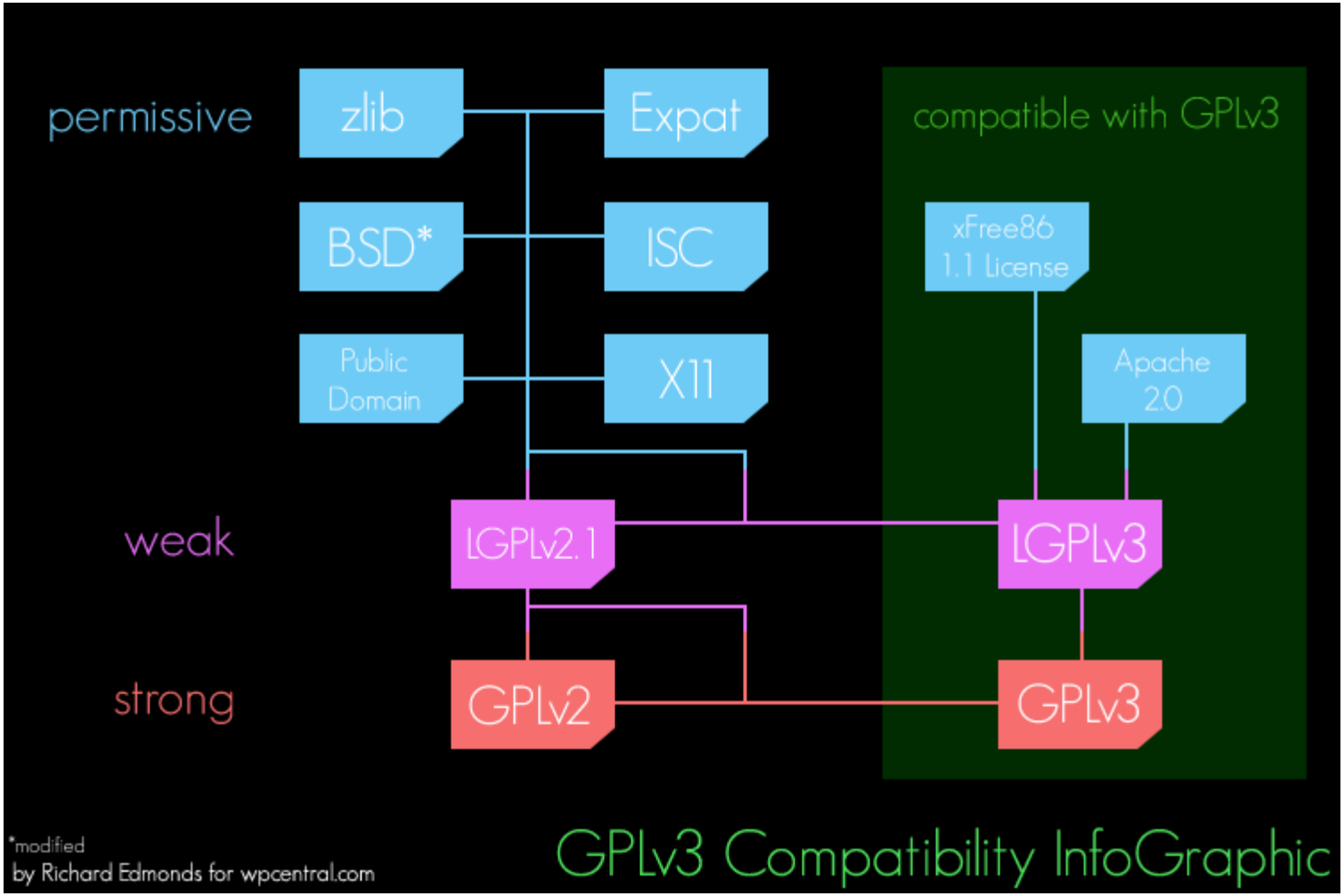
SonarQube – Spazio IT

- But… For contractual and «IP» reasons the plugin developed by Spazio IT for AIRBUS Helicopters could not be given back to the community.

- Spazio IT is currently refactoring its plugin so that it can be divided into a core module (open source) and an extended one (proprietary).

Spazio IT – SonarQube - Demo

Licenses and IP



Licenses and IP

- Open source software can be adopted in avionics if and only if its license does not impose any condition whatsoever on the software system built with it.

Qualification and Certification

Qualification and Certification

- Qualification: The process of demonstrating the ability to fulfil specified requirements. Note: the term ‘qualified’ is used to designate the corresponding status. [ISO 9000]

- Certification: The process of confirming that a component, system or person complies with its specified requirements, e.g. by passing an exam.
  - E.g. The X Agency confirms that Software Product Y has been qualified at DO-178(B)C DAL B level

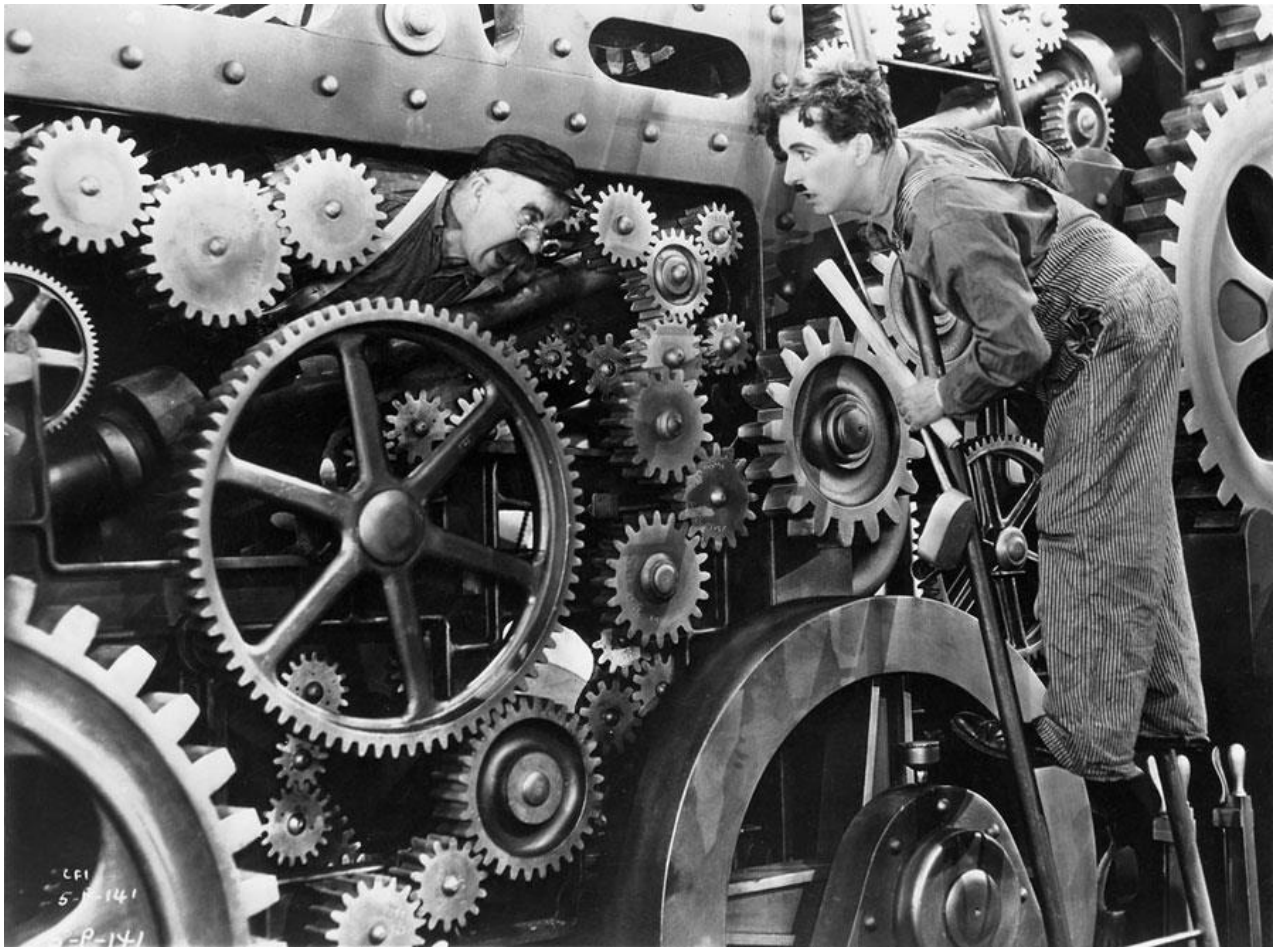
Qualification and Certification

- Qualification and certification are very expensive... Who is going to actually doing them?

- The open source community? No!

- Who then?
  - the end user of the software;
  - a group, an alliance of users;
  - a specialised company, anyhow paid by a user or a group of users.
    (e.g. Spazio IT can verify the «MISRA Compliance» of software written in C/C++).

Maintenance and Support



Maintenance and Support

- Avionics systems have an «operational lifetime» varying **from 3/5 to 15/20 years**. Development time alone can also vary from 3/5 to 10 years.

- Is there any so long-lived open source community?

Maintenance and Support

- Who can actually offer maintenance and support for such a long period? Once again:
  - the end user of the software;
  - a group, an alliance of users;
  - a specialised company, anyhow paid by a user or a group of users.

- **NOTE**: in avionics the only interesting versions are the «LTS» ones.

Economic Sustainability

Economic Sustainability

- The open source business model (for quite some OSS) which distinguishes between:
  - (free) «Community Edition»
  - (paid) «Enterprise Edition»

  and relies on support as continuous source of income is not really applicable to avionics software, given the small number of (potential) customers and the fact that these customers have different needs.

- A model which is often adopted:
  - 1 «Community Edition»
  - N «Enterprise Editions» (one per customer)

Economic Sustainability

- When talking about qualification and maintenance, three different entities:
  - the actual end user of the open source software;
  - a group, an alliance of end users;
  - a specialised company, anyhow paid by a user or a group of users.
- Usually of these three entities only the fist two are «strong» enough to support the software for a long time...
- So the could act as «sponsors», e.g.:
  - DOD → OAR
  - ESA → RTEMS Centre

Thank you for your attention!