MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES
POR LA UNIVERSIDAD POLITÉCNICA DE MADRID

POLITÉCNICA

2021.02.08

Data handling in space missions

Juan Antonio de la Puente <juan.de.la.puente@upm.es>
Alejandro Alonso <alejandro.alonso@upm.es>

STRAST

Outline

- Data handling in space missions
  - on-board and ground segments
- Computer architecture
  - generic system architecture
  - hardware and software architecture
- Requirements, limitations and design process

Data handling in space missions

- Data handling is crucial in space missions
- Most aspects of the mission are controlled by computers & software
  - launch trajectory & orbit acquisition
  - attitude control
  - platform control
  - telecommunications
- Missions may fail because of hardware/software faults
  - data handling is a *mission critical* component

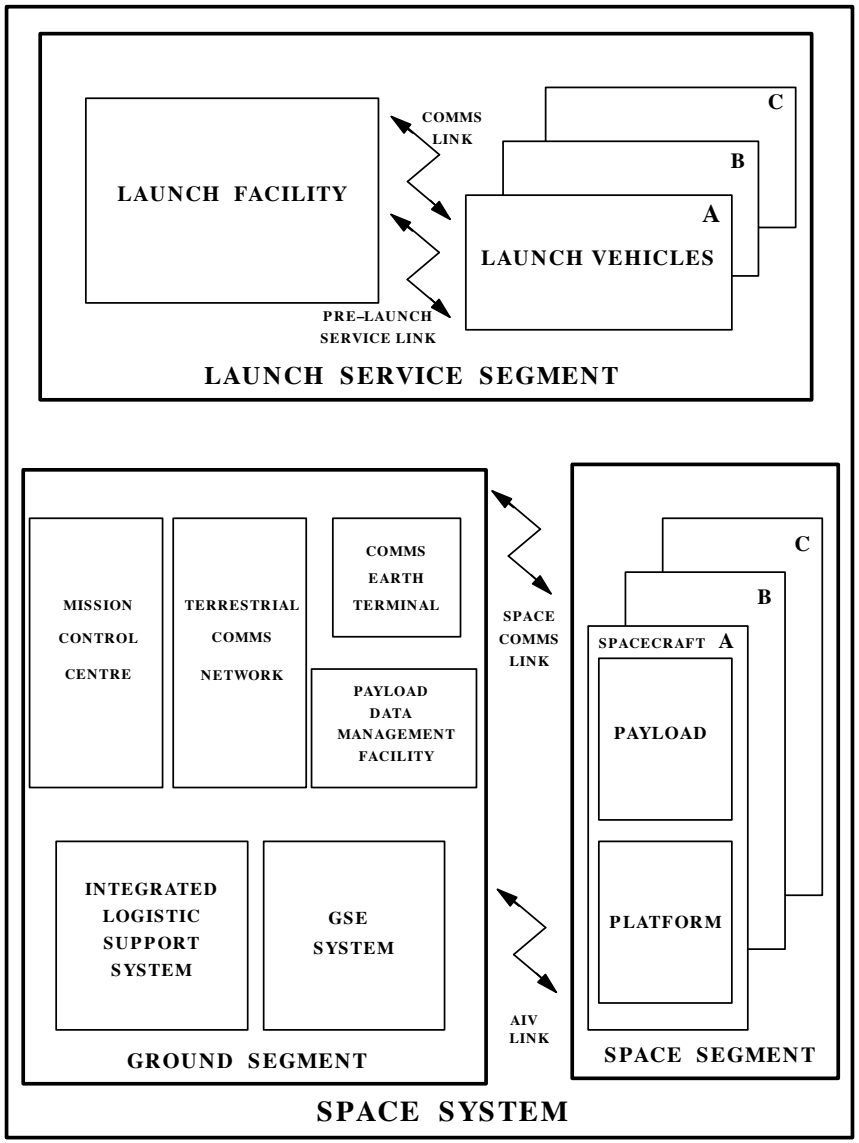
Some notorious software failures

- Ariane 5 launcher (1996)
  - self-destroyed because of a failure in flight control software
  - incorrect reuse of previous (Ariane 4) software
  - correct data misinterpreted as erroneous
- Mars Pathfinder (1997)
  - repeated timing failures led to frequent resets
  - bad design of real-time parameters
  - fixed by uploading a software patch
- Mars Climate Orbiter (1999)
  - persistent navigation errors resulted in entering Mars atmosphere at an improperly low altitude and probably burning up
  - mixing SI and imperial units went undetected

… and a few more ones in

- http://www.itworld.com/article/2823083/88716-8-famous-software-bugs-in-space
- https://en.wikipedia.org/wiki/List_of_software_bugs#Space

Space system segments



Source: ECSS-E-00A

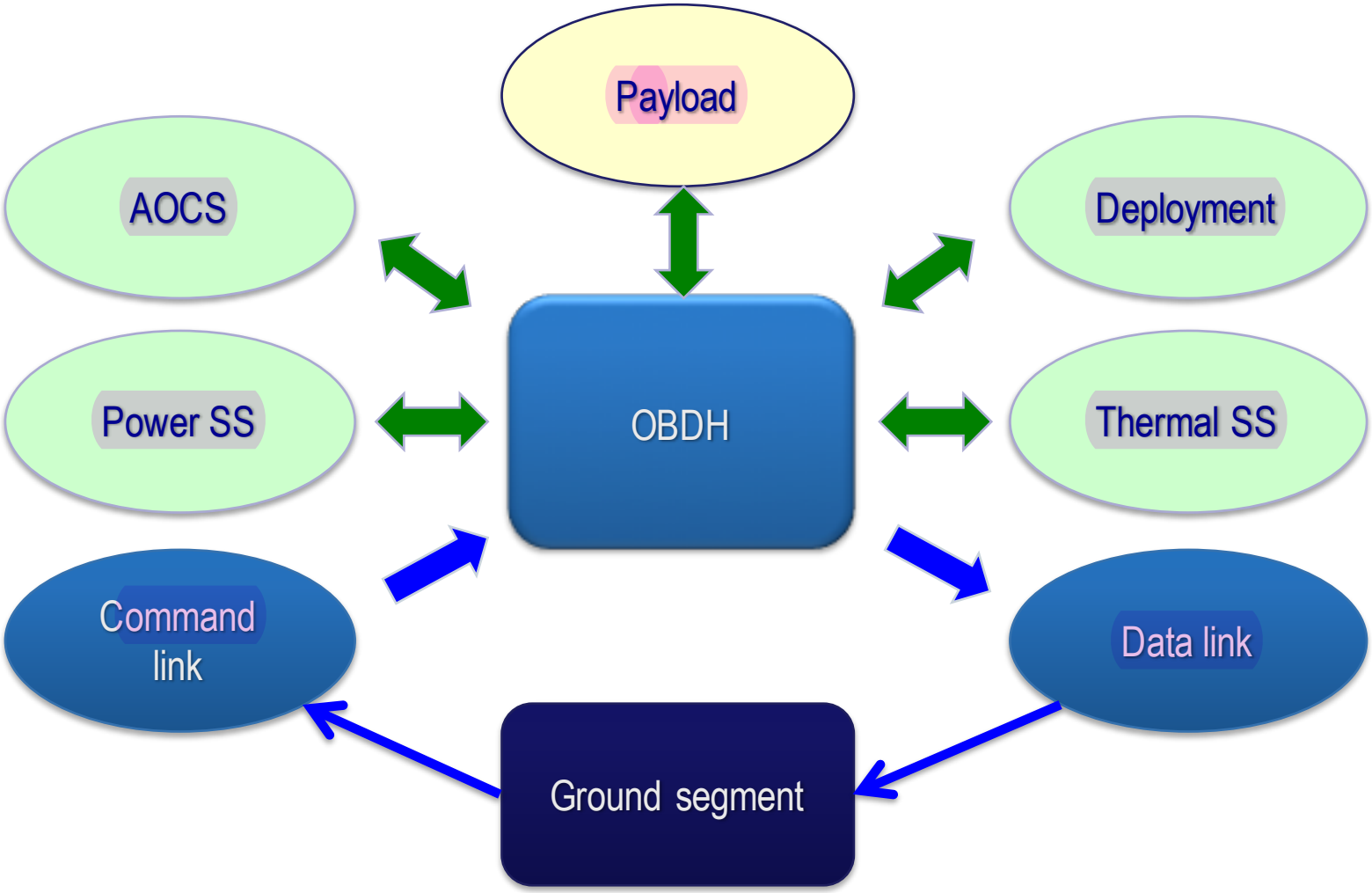
NOTE GSE = Ground Support Equipment
AIV = Assembly, Integration, Verification

- The launch and space segments have the most demanding requirements for
  - reliability,
  - safety
  - integrity
- On-board computers are subject to harsh environmental conditions
- On-board hardware/software cannot be replaced if defective

On-board Data Handling (OBDH)

- The OBDH subsystem is in charge of exchanging data between the spacecraft functional units and the ground segment
  - data storage and distribution within the spacecraft as well
- It is highly integrated with the TT&C subsystem
  - telemetry, tracking and telecommand
- It is based on one or more on-board computers (OBC)

OBDH architecture

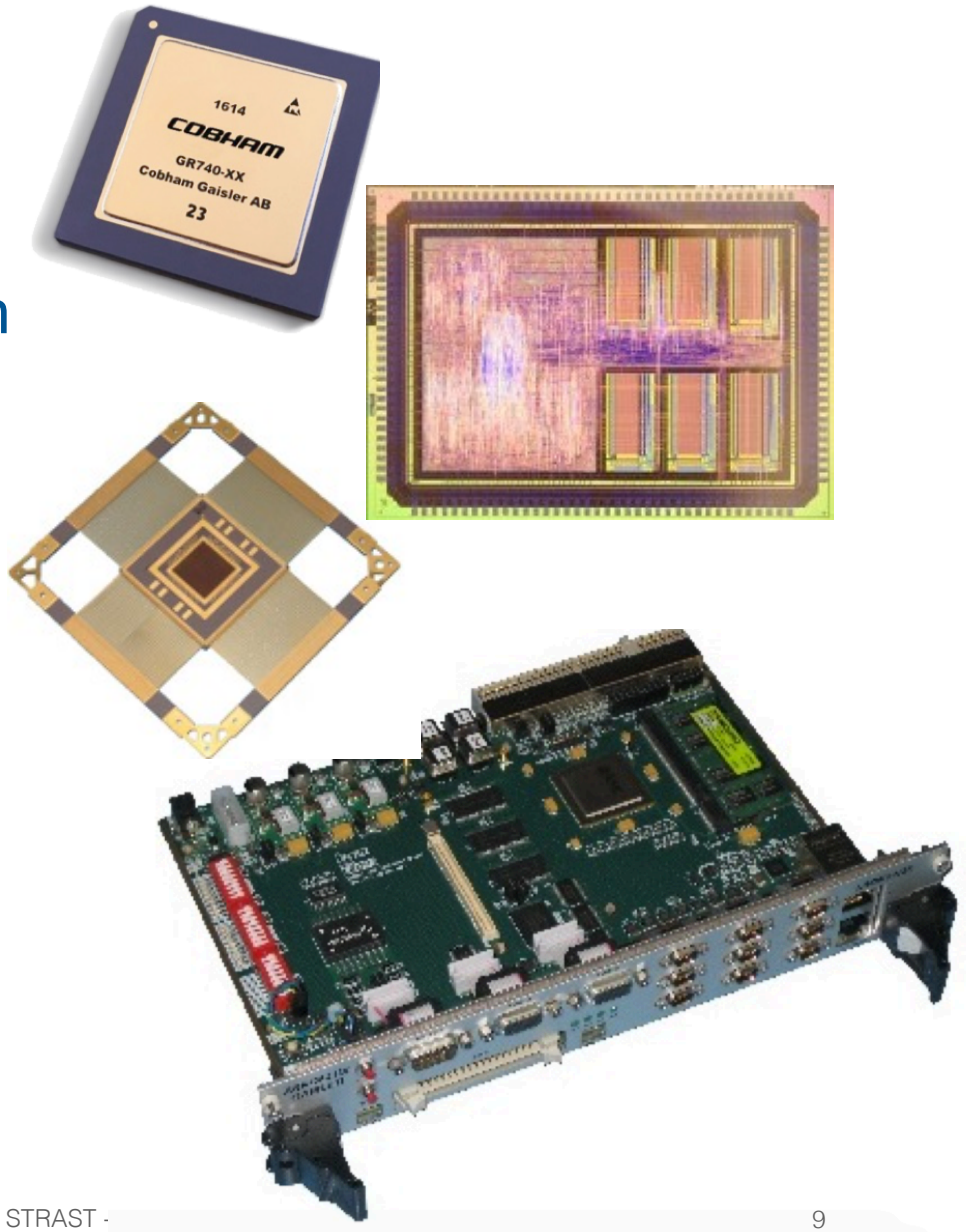


On-board computers

- Computation support to spacecraft functions
- Specific requirements
  - embedded in spacecraft
    - specific I/O devices, limited power, size and weight
  - real-time operation
    - functions have to be performed on time
  - high reliability requirements
    - repair is not possible in space
  - harsh environment
    - radiation, temperature, vibration, acceleration
- Applicable technology may be restricted
  - often resulting in less powerful technology being used

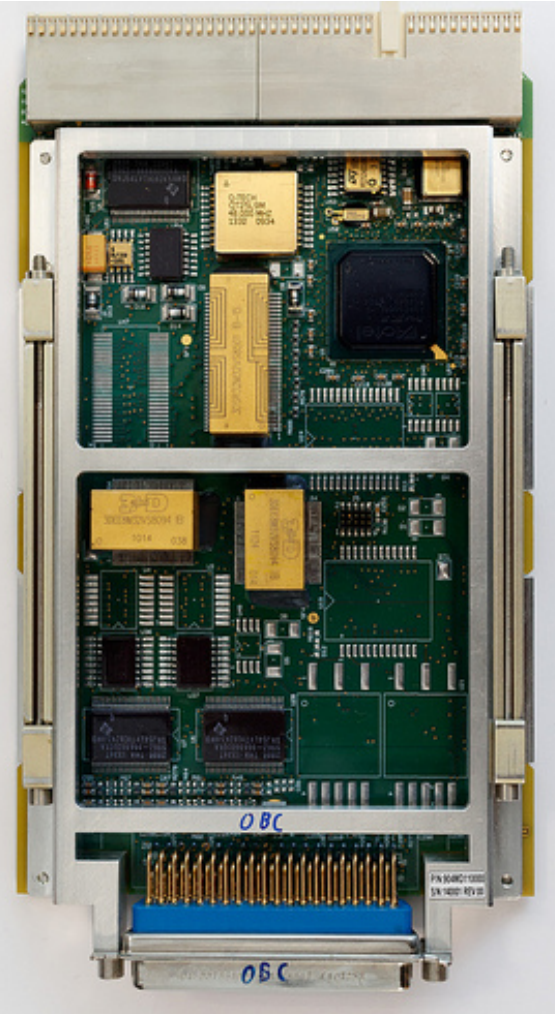
OBC hardware

- LEON processors are common in ESA systems
  - open-source µP core from which chips can be built
- The current version is LEON4
  - SPARC v8 compliant RISC architecture
  - radiation-hard implementations
  - speed ≈ 100 MHz
  - power ≈ 0,5 W



https://en.wikipedia.org/wiki/LEON

Example: UPMSat2 on-board computer

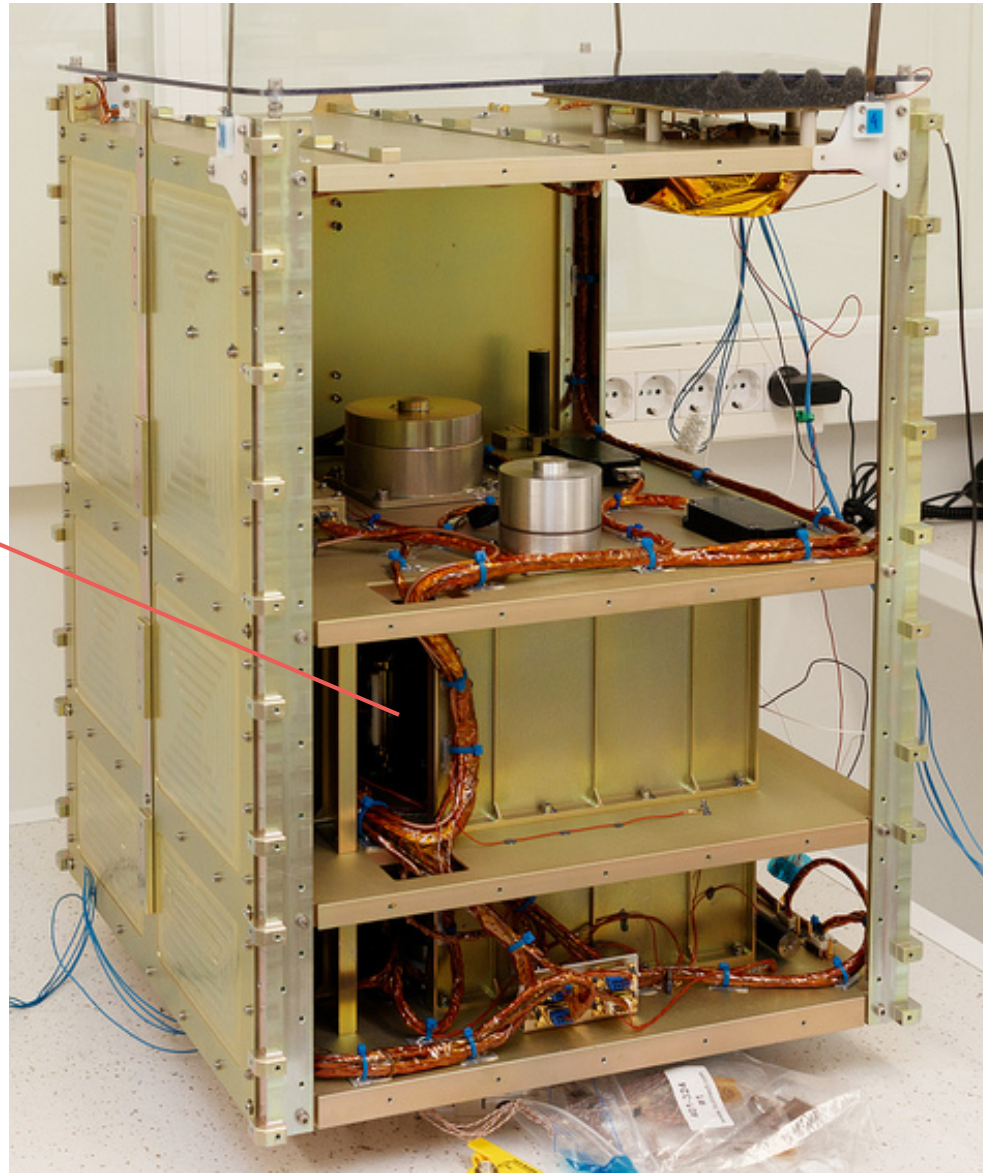


Computer board based on LEON3

EBOX hosting the computer board

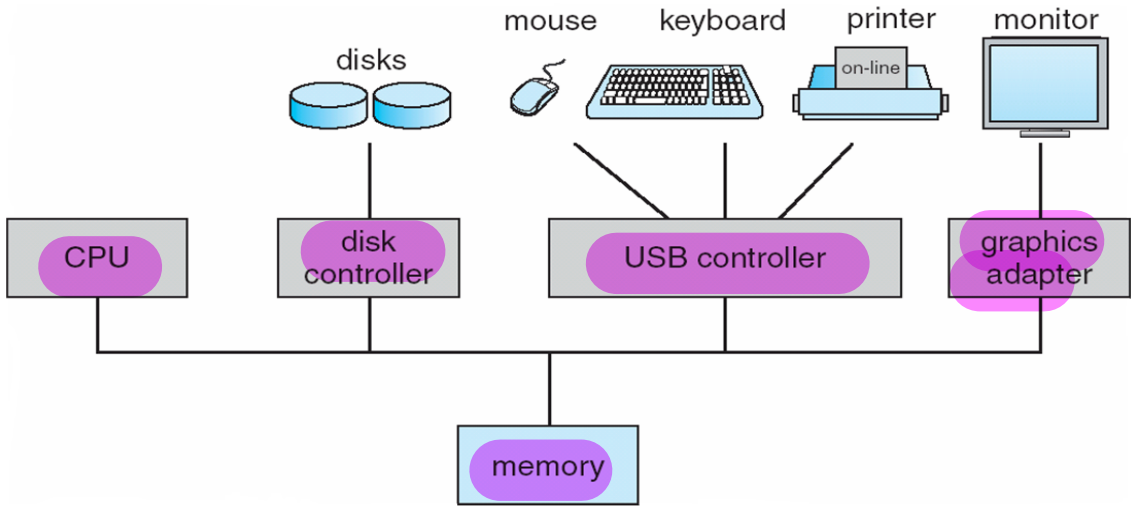
http://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Onboard_Computer_and_Data_Handling2

EBOX

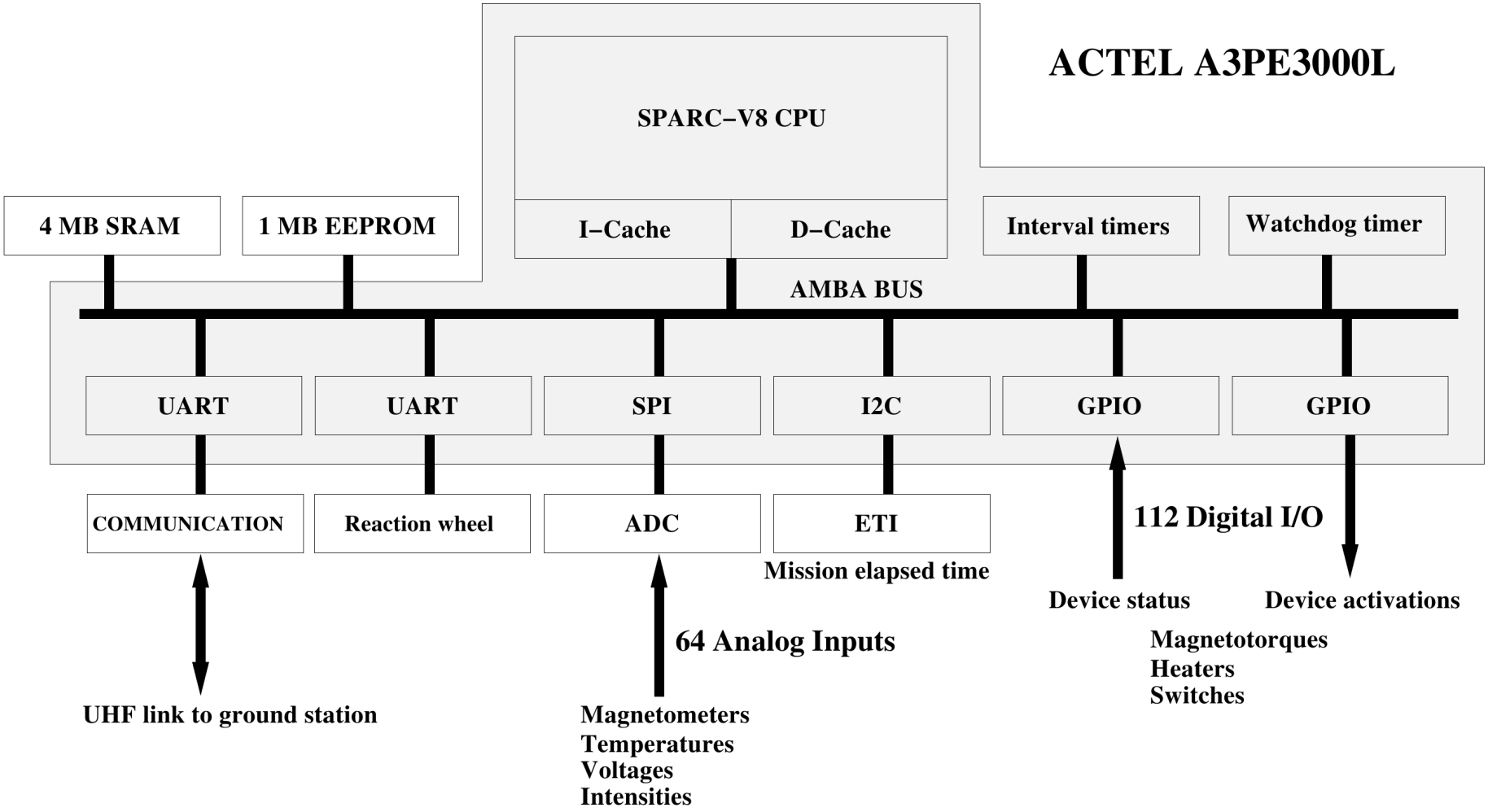


Computador de Von Neumann

- Modelo de computador propuesto en 1947
  - El más empleado.
  - Hay otras alternativas (procesadores de gráficos, FPU)
- Características
  - Datos e instrucciones almacenados en memoria
  - Contenido de la memoria accesible por direcciones
  - Ejecución implícitamente en secuencia



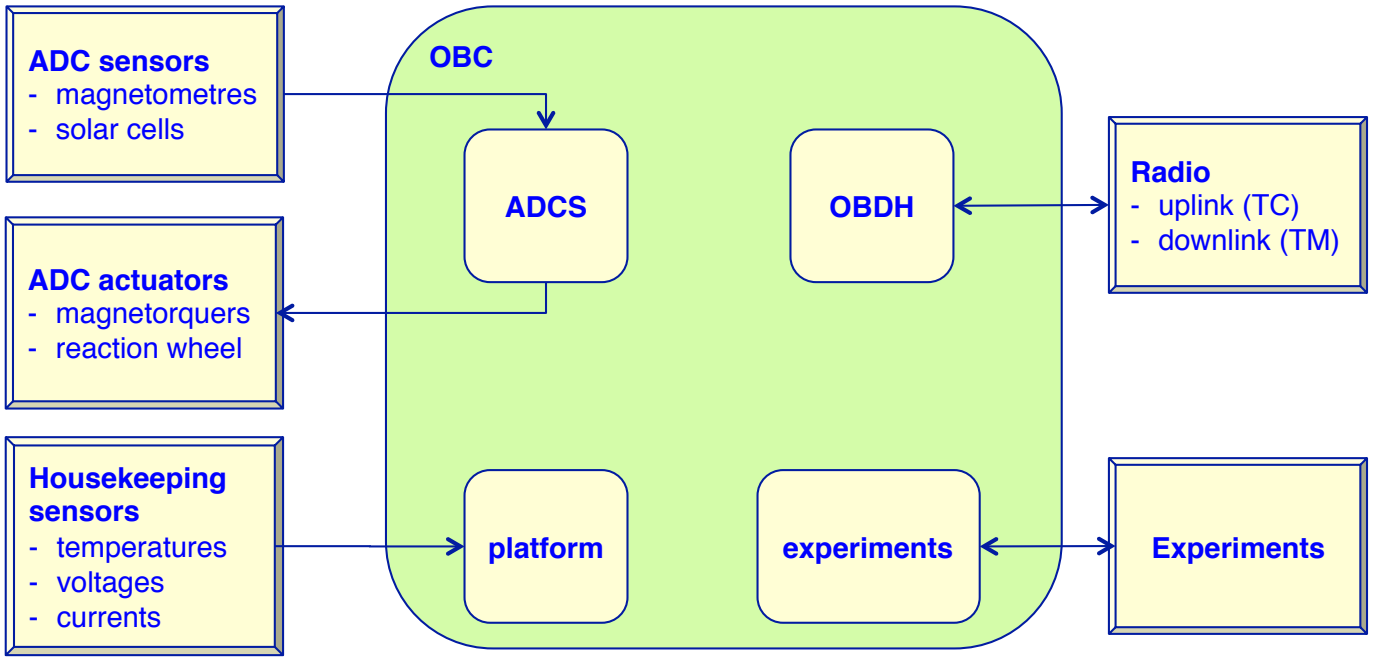
UPMSat2 OBC hardware



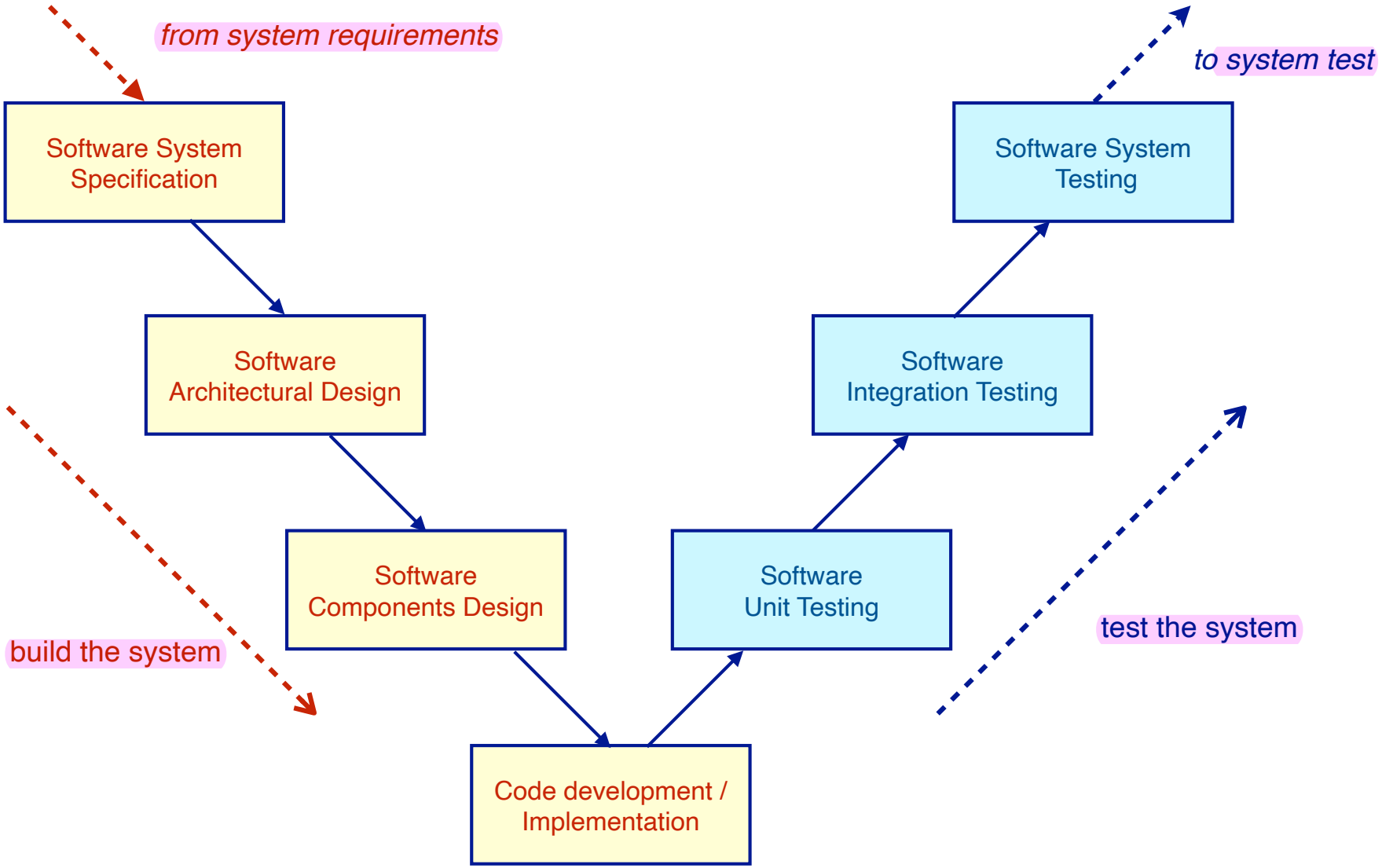
OBC software

- Many critical functions depend on software
  - high-integrity requirements
- Most software functions have real-time requirements
  - do things in time
- Verification & validation process is crucial
  - safety management
- Technology choice driven by high-integrity requirements
  - e.g. Ada, RTOS, static analysis, temporal analysis

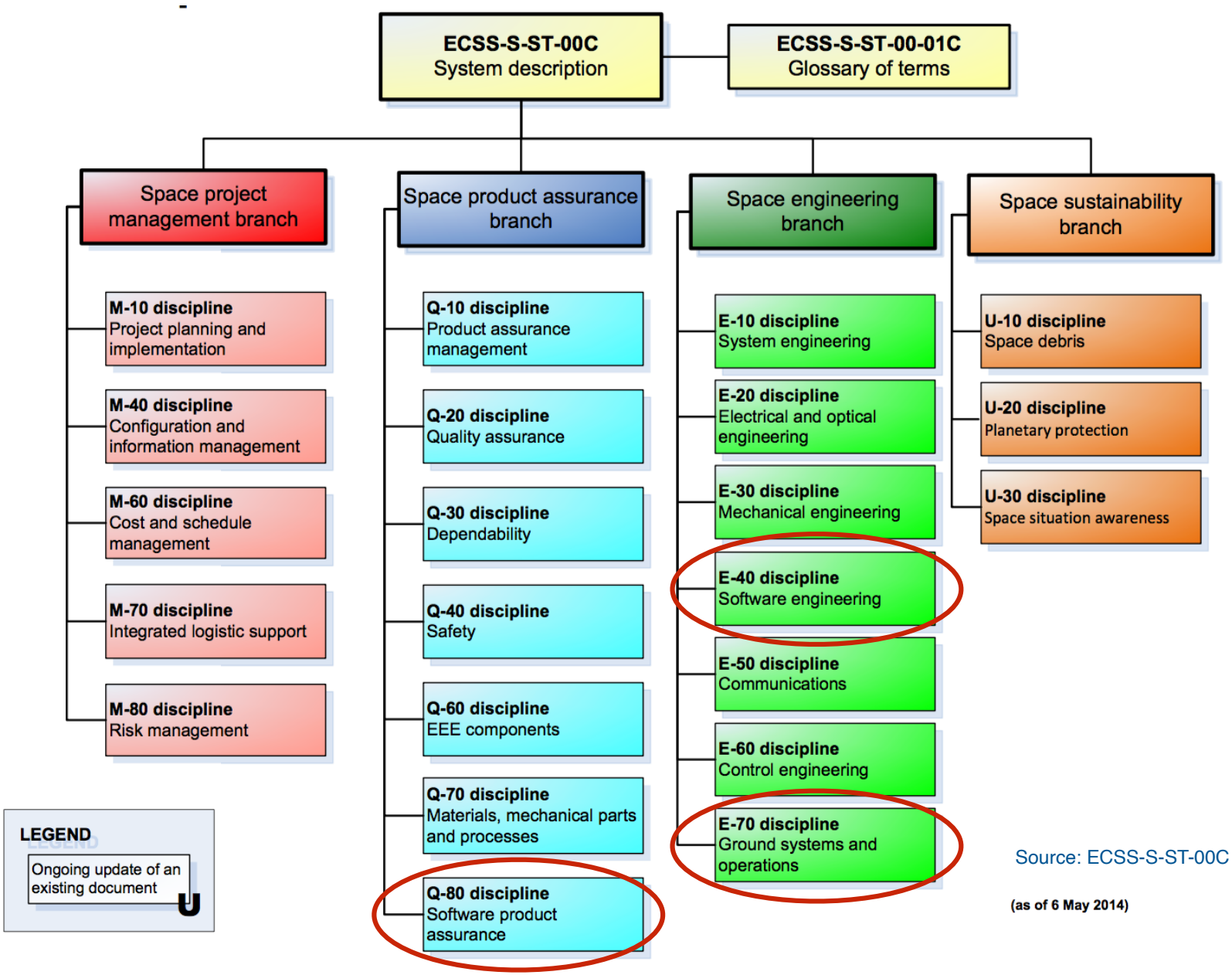
UPMSat2 software



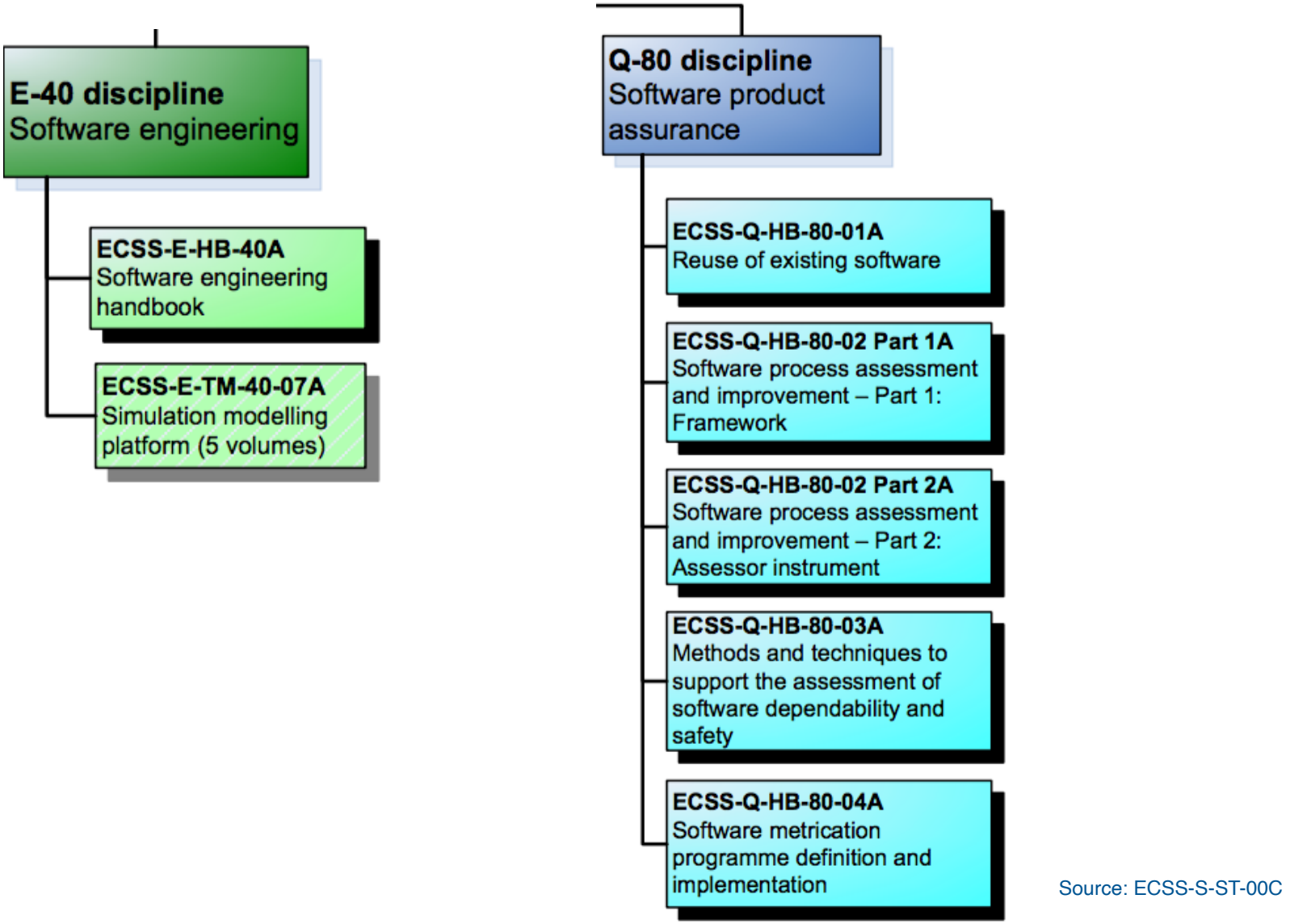
Software development cycle



ECSS standard disciplines

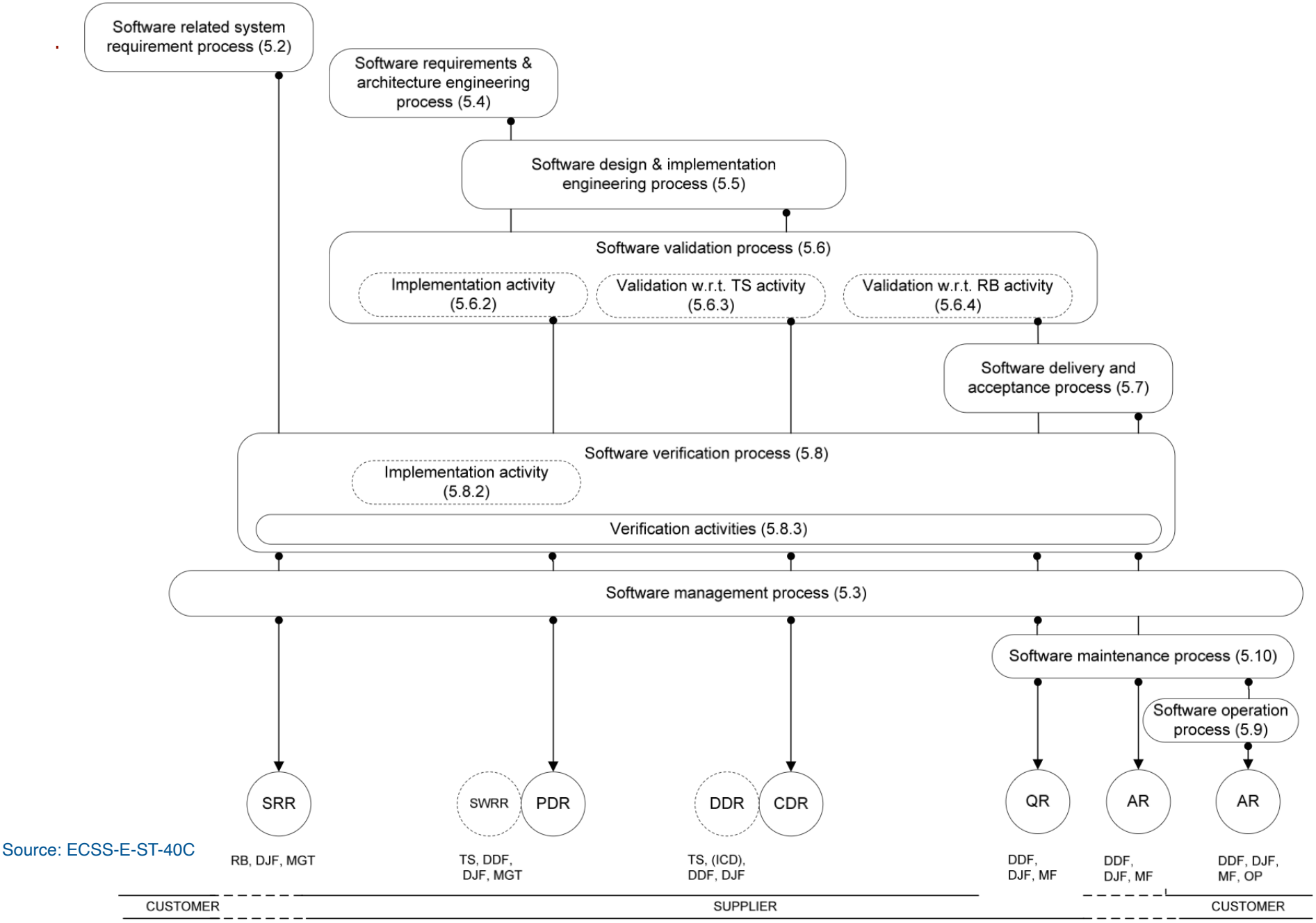


ECSS handbooks



Source: ECSS-S-ST-00C

ECSS software processes



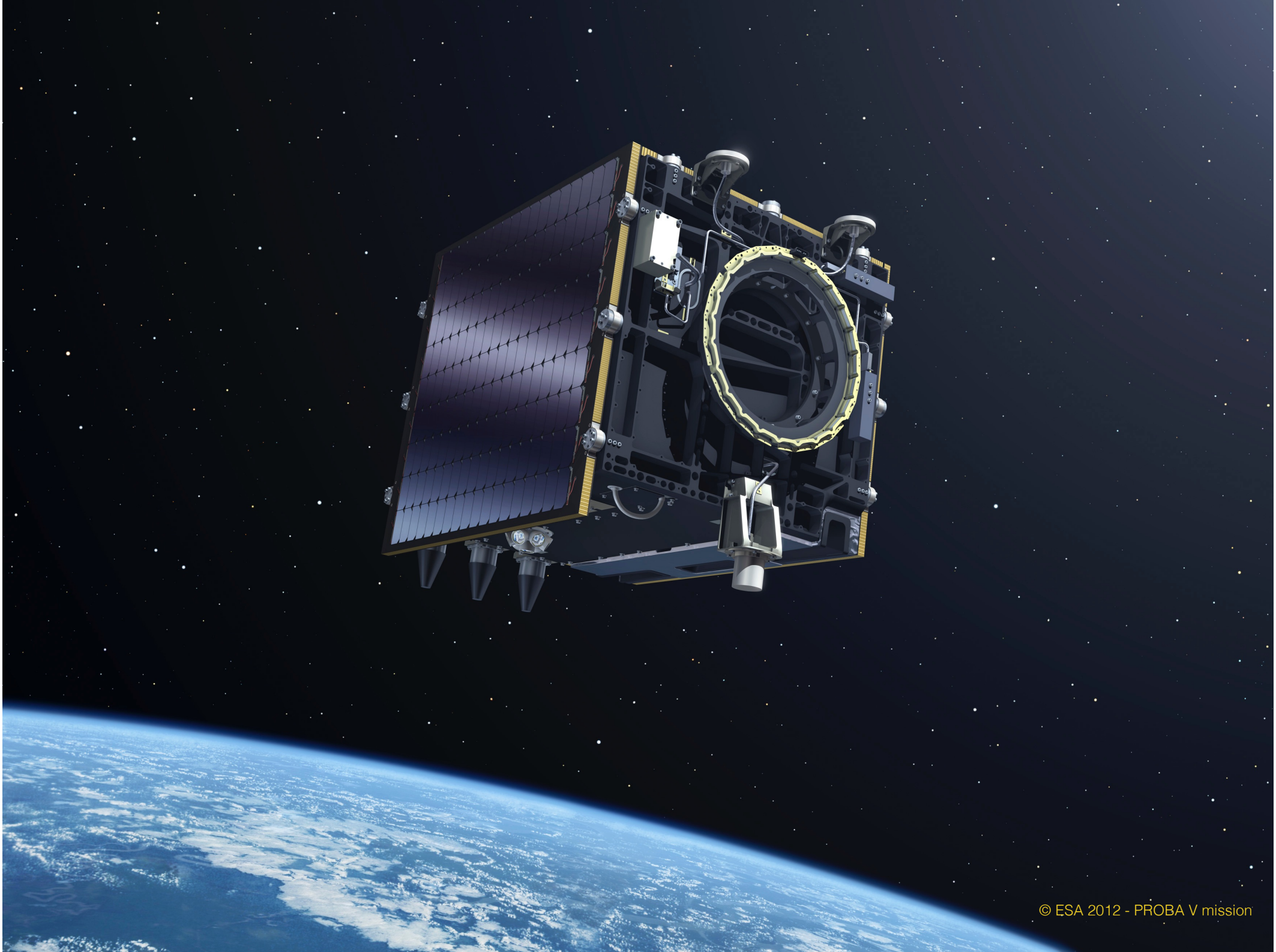
Source: ECSS-E-ST-40C

ECSS software criticality categories

Category	Definition
A	Software that if not executed, or if not correctly executed, or whose anomalous behaviour can cause or contribute to a system failure resulting in: **Catastrophic consequences.**
B	Software that if not executed, or if not correctly executed, or whose anomalous behaviour can cause or contribute to a system failure resulting in: **Critical consequences.**
C	Software that if not executed, or if not correctly executed, or whose anomalous behaviour can cause or contribute to a system failure resulting in: **Major consequences.**
D	Software that if not executed, or if not correctly executed, or whose anomalous behaviour can cause or contribute to a system failure resulting in: **Minor or Negligible consequences.**

Summary

- Data handling is a critical component of space systems
- On-board computer systems are most critical
- OBC hardware design is driven by reliability requirements in a harsh environment
- OBC software design is driven by high-integrity requirements
- ECSS standards for software define processes for developing high-integrity software
  - ECSS-E-40C
  - ECSS-Q-80C


© ESA 2012 - PROBA V mission