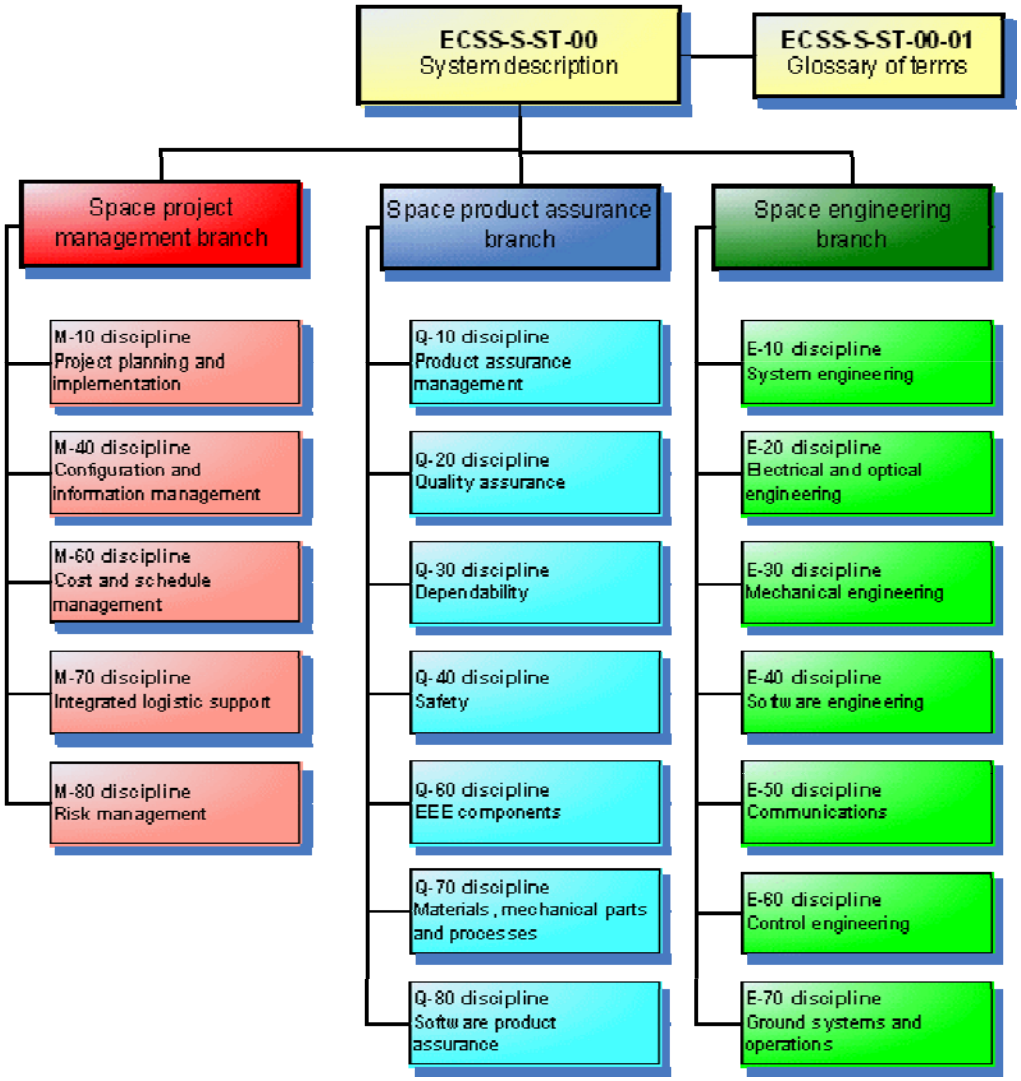
Mission Operation Ground Software Systems Product Assurance @ ESA

Mario Merri
GSAW, Los Angeles, USA
2 Mar 2011

The European Cooperation for Space Standardisation (ECSS)

- Established: in 1993
- Goal: coherent, single set of standards for use in all design and development European space activities
- Coverage: space and ground segments
- Membership: European Space Agencies and European Industry
- Website: www.ecss.nl

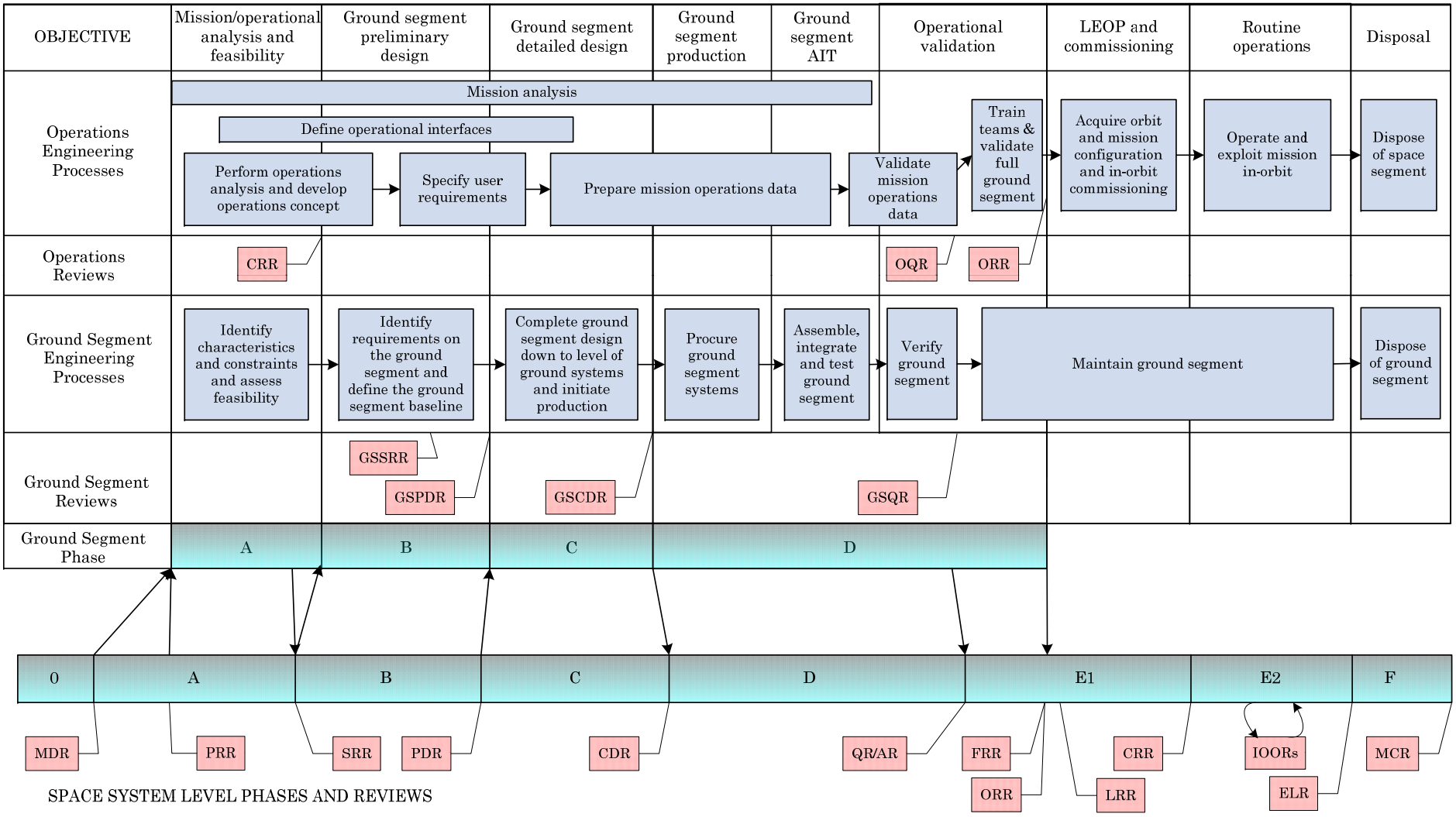


Tailoring of ECSS Standard

1. ECSS foresees tailoring of standards

"Tailoring is a process by which individual requirements or specifications, Standards and related documents are evaluated and made applicable to a specific project, by selection and in some exceptional cases, modification of existing or addition of new requirements"

ECSS-E-ST-70: Ground Segment and Operations Phases



Relevant Standards for Ground Software Systems Quality Assurance

1. ECSS-E-ST-40 “Space engineering — Software”

2. ECSS-Q-ST-80 “Space product assurance — Software product assurance”

Other related standards

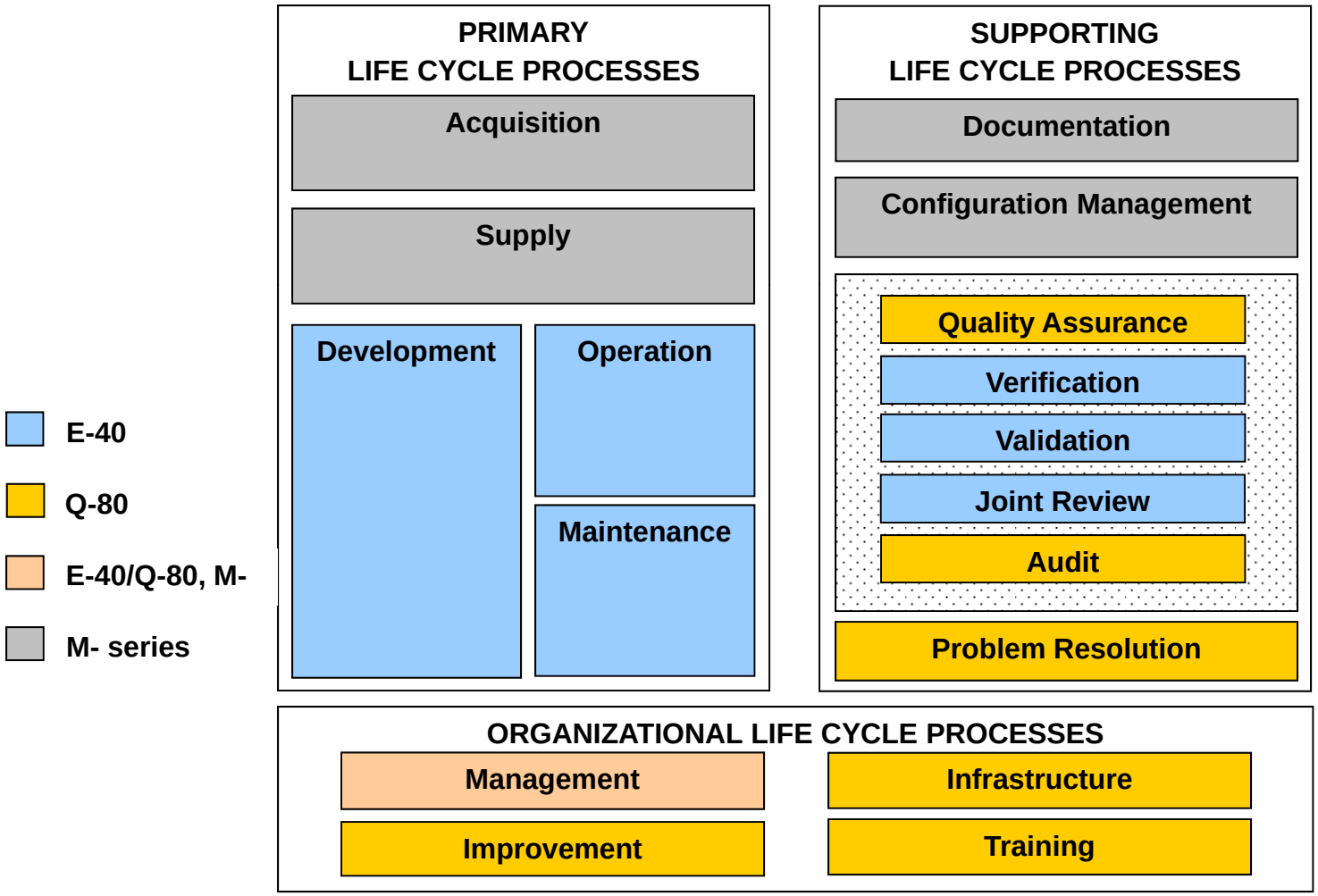
a. ECSS-Q-ST-10 “Space product assurance - Product assurance management”
   - to ensure that space products accomplish their defined mission objectives in a safe, available and reliable way

b. ECSS-Q-ST-20 “Space product assurance — Quality assurance”
   - to ensure that a QA programme for projects covering mission definition, design, development and production of space systems is established, maintained and implemented

c. ECSS-Q-ST-30 “Space product assurance — Dependability”
   - identification of all technical risks
   - application of analysis/design methods to ensure that dependability targets are met,
   - optimization of the overall cost and schedule by making sure that:
     - introducing appropriate severity categorisation
     - risks reducing actions are implemented continuously

d. ECSS-Q-ST-40 “Space product assurance — Safety”
   - to ensure that all safety risks associated with the design, development, production and operations of space products are adequately identified, assessed, minimized, controlled and finally accepted

Software Related Processes in ECSS Standards



ECSS-E-ST-40 “Space Engineering - Software”

1. Defines the space software engineering life cycle and processes and their expected outputs
    a. SW requirements and architecture engineering process
    b. SW design and implementation engineering process
    c. SW validation process
    d. SW delivery and acceptance process
    e. SW verification process
    f. SW operation process
    g. SW maintenance process

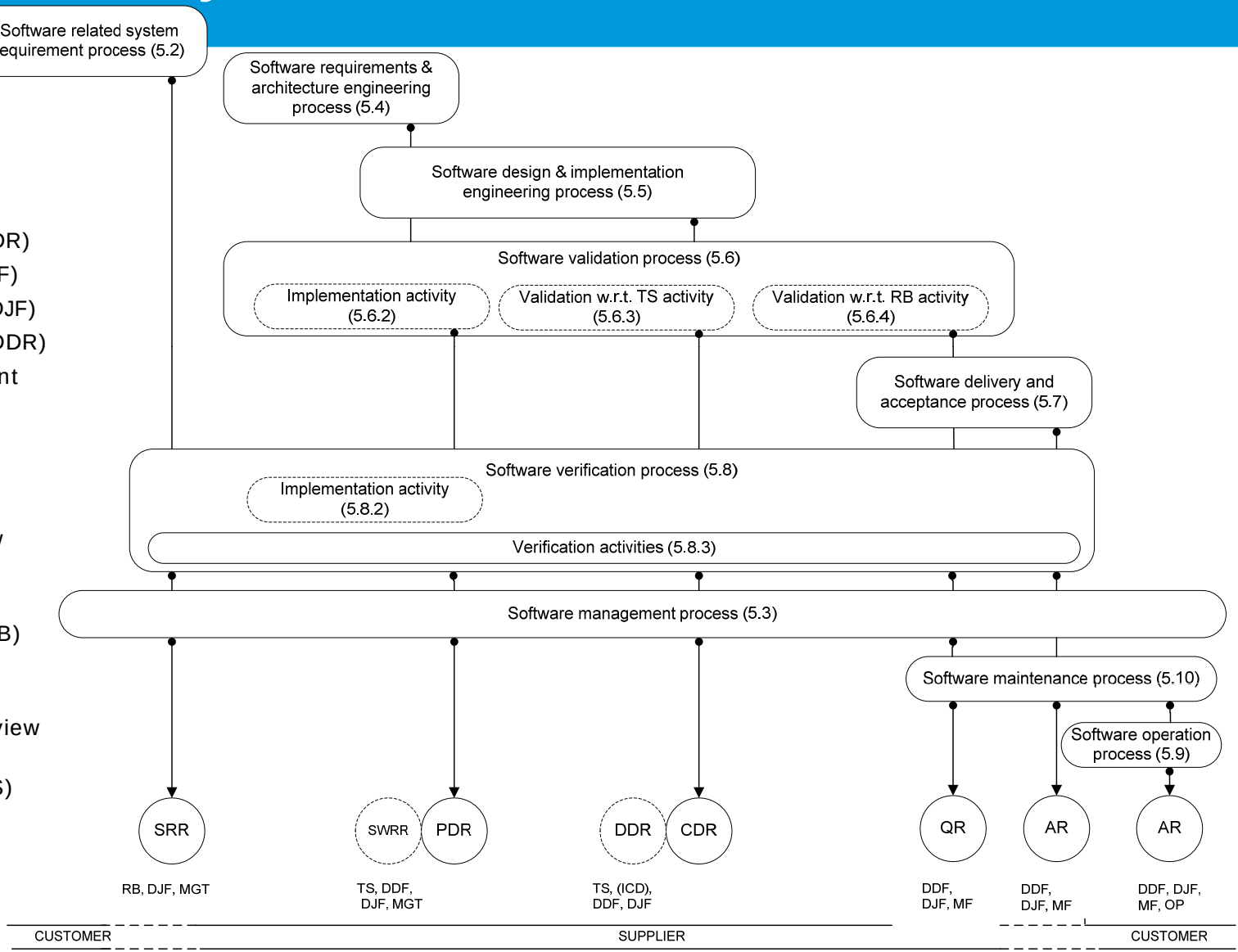
2. Associates reviews to each process

3. Covers both flight and ground SW

4. Defines the required documentation and their content

ECSS-E-ST-40: Overview of the Software Lifecycle Process

Acceptance Review (AR)
Critical Design Review (CDR)
Design Definition File (DDF)
Design Justification File (DJF)
Detailed Design Review (DDR)
Interface Control Document (ICD)
Maintenance File (MF)
Management File (MTG)
Operational Plan (OP)
Preliminary Design Review (PDR)
Qualification Review (QR)
Requirements Baseline (RB)
Software Requirements Review (SWRR)
System Requirements Review (SRR)
Technical specification (TS)



ECSS-Q-ST-80 “Space Product Assurance - Software Product Assurance”

1. Defines a set of software product assurance requirements applicable to the development and maintenance of software for space systems
2. Software product assurance programme implementation
    a. Organization responsibility, and programme management
    b. Risk management and critical item control
    c. Supplier selection/control, Procurement
    d. Tools and supporting environment
    e. Assessment and improvement process
3. Software process assurance
    a. Software development life cycle
    b. Requirements applicable to SW engineering processes
4. Software product quality assurance
    a. Product quality objectives, metrication and requirements
    b. Software intended for reuse
    c. Standard ground hardware and services for operational system
    d. Firmware

Peculiarity of Ground Software Systems

1. High complexity of the overall ground segment (many complex systems working in different configurations)
2. Use of third party products in many areas (operating system, software, hardware, firmware, e.g. SW running on routers)
3. Contain non space-specific elements/functionality e.g. Middleware, Data management
4. At ESA, normally based on an existing infrastructure. Mission specific development must conform to it (e.g. compatibility with ICT infrastructure)
    a. Software Infrastructure (reusable software) is essential for provision of mission dedicated systems at reduced cost and risk
5. Ground systems are accessible and can be corrected and redeployed while complying with availability requirements
6. In addition to V&V through software engineering practice, also validated in operations engineering life cycle, e.g. operational validation and simulations campaign

Impact and Direct Applicability of E40 & Q80

1. Pre-tailoring by criticality & dependability analysis (E40)

2. Code coverage (E40)

Code coverage versus criticality category	A	B	C	D
Source code statement coverage	100%	100%	AM	AM
Source code decision coverage	100%	100%	AM	AM
Source code modified condition and decision coverage	100%	AM	AM	AM
NOTE: "AM" means that the value is agreed with the customer and measured as per ECSS-Q-ST-80 clause 6.3.5.2.				

3. Usage of third party products (Q80)

Pre-tailoring by Criticality & Dependability Analysis (1)

1. ECSS requires that **functions** and **products** be classified based on their criticality level

2. The criticality of a **function** is directly related to the severity of the **consequences** of the most severe failure of that function

3. The criticality of a **product** (HW and SW) is the highest of the criticality levels among the functions associated with that product

Category	*Definition*
A	**Software that if not executed, or if not correctly executed, or whose anomalous behaviour could 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 could 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 could 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 could cause or contribute to a system failure resulting in:** • **Minor or Negligible consequences**

Pre-tailoring by Criticality & Dependability Analysis (2)

1. SW criticality & dependability analysis (e.g. Software Failure Modes, Effects and Criticality Analysis - SFMECA) can only be applied in a context or environment (e.g. mission):
    a. Not applicable to infrastructure SW
2. The SFMECA does not consider
    a. Likelihood of failures
    b. Continuity aspects (e.g. how long the system can be unavailable)
    c. Fault prevention measures for criticality evaluation
    d. Operator errors

Available pre-tailoring is not sufficiently discriminatory for Ground Software Systems

Code Coverage (1)

1. E40 puts strong emphasis on code coverage
    a. Code coverage analysis is useful, but only a contribution to testing
2. Large number of possible configurations in infrastructure SW
    a. Each configuration allows different areas of the code to be covered
    b. It is impossible to cover all these configurations and associated areas of code
3. Many ground software systems are multithreaded
    a. Test of multithreaded systems is best done by system testing in the target environment
4. System validation campaigns (main validation exercise) cannot be performed with instrumented code
    a. Instrumented code cannot carry credits for system validation

Code Coverage (2)

1. **S.L. Pfleeger, L. Hatton ,C. Howell, Solid Software, Software Quality Institute Series, Prentice Hall PTR, 2002**
    a. Ch. 4. *Perform unit testing with 100 percent coverage … However, … it wastes resources to retest code components once they become stable. And it is just as wasteful to perform tests without considering the utility of each component … **just because you've covered the code under test doesn't mean that it does anything useful or that it even works** …*
    b. Ch. 4. ***Test the entire system end to end. Testing the system end to end enables testers to find problems that are not evident in individual components nor in their interfaces***.
    c. Ch. 4. *Use an operational profile and testing to predict the expected failure rate of the software under a given load*.
2. **Andrew Glover. In pursuit of code quality: Don't be fooled by the coverage report** http://www-128.ibm.com/developerworks/java/library/j-cq01316/index.html
    a. *High coverage rates simply mean that a lot of code was exercised. **High coverage rates do not imply that code was exercised well**. If you're focusing on code quality, you need to understand exactly how test coverage tools work, as well as how they don't; then you'll know how to use these tools to obtain valuable information, rather than just settling for high coverage goals, as many developers do.*

Third Party Products (1)

1. The Ground Systems have many millions of 3rd PP LoCs
2. 3rd PP engineering details typically not available to users
3. 3rd PP service history not available at all or not relevant for reuse scenario (e.g. never been used in satellite operations)
4. Versioning of SW product "reset" service history
5. Some 3rd PP also use 3rd PP
6. Even for small 3rd PP the effort implied by Q80 Clause 6.2.7 can be close to impossible

Extract from Q80...

6.2.7.8

a. Reverse engineering techniques shall be applied to generate missing documentation and to reach the required verification and validation coverage.

b. For software products whose life cycle data from previous development are not available and reverse engineering techniques are not fully applicable, the following methods shall be applied:

1. generation of validation and verification documents based on the available user documentation (e.g. user manual) and execution of tests in order to achieve the required level of test coverage;

2. use of the product service history to provide evidence of the product's suitability for the current application, including information about:

(a) relevance of the product service history for the new operational environment;

(b) configuration management and change control of the software product;

(c) effectiveness of problem reporting;

(d) actual error rates and maintenance records;

(e) impact of modifications.

EXPECTED OUTPUT: Software reuse file [DJF, SRF; SRR, PDR].

Third Party Products (2)

1. Ground Segment Subsystem and System level validation campaigns are fundamental to put the reused product in the right context
2. 3rd PP are validated as a complete stack as part of system validation in the same conditions and environment where they will be used
3. Special attention paid to open source (enhanced quality due to “network effect” (Gartner RP ID G00144771))
4. Similar approaches used for safety critical applications in other areas
    a. RTCA/DO-278: third party SW “assurance credit” may be based on system validation process, pre-operational activities and operator training

Ground Software Systems require special tailoring of ECSS-E-ST-40 and ECSS-Q-ST-80 and different criteria for criticality analysis

Addressing the Concerns

In 2008, two initiatives were started which are now completed:

1. An ESOC WG was created with the mandate to produce a suitable tailoring of ECSS-E-ST-40C and ECSS-Q-ST-80C to be applied for Ground Software Systems at ESOC

2. A SW Criticality Analysis study was performed on the GAIA Mission Control System (MCS) with the objective of
    a. Identifying criticality level of MCS SW functions
    b. Generalising of results for different Ground Software Systems

Tailoring of E40 & Q80 for Ground Software Systems

Four tailoring cases were identified:

1. **TT1: Full Lifecycle Developments**
2. **TT2: High Degree of Re-use of Existing Infrastructure**
    a. e.g. mission control systems and operational spacecraft simulators
3. **TT3 : Evolution of Existing Infrastructure**
    a. This type of project is not directly deployed in a specific ground segment, e.g. SCOS-2000, SimSat
4. **TT4 : Prototype and Study Software Projects**
    a. minimal requirements for documentation and validation
    b. not intended for reuse

This tailoring was agreed at ESA level and documented in ESOC Quality Management System

SW Criticality Analysis Study

1. Approach
    a. Development of top down analyses (System level) to identify the project relevant Feared Events and their severity categories
    b. Identification of critical functions
    c. Definition of the critical subsystems and of critical items and components depending on their functionalities
    d. Identification of the most appropriate design solutions to prevent events occurrence or to mitigate the consequences

2. Conclusions
    a. The study confirmed that the ECSS SW criticality classification is too rigid as
        – it is exclusively based on the severity of consequences in case the SW is not correctly executed
        – with no consideration of existing compensating provisions (e.g. SW or HW back-up, operator surveillance, partitioning)!

Next Step in SW Criticality Analysis

1. A working group has been created to
    a. Establish an approach to allow the determination of the software criticality category taking into account prevention/mitigation mechanisms
    b. Provide clear guidance about the legitimate conditions for software criticality downgrading

... stay tuned