Software Product Assurance

Manrico Fedi Casas – Head of SW PA at ESA/ESTEC

About me

- Head of the SW Product Assurance section in ESTEC (TEC-QQS)
- Previously SW Product Assurance engineer.
- Supported projects in different directorates (HRE, SCI, EOP, TIA).
- Background as RAMS engineer and SW engineer.
- Ph.D. Aerospace Engineering.

Software Product Assurance

- ❑ Block #1 – Software in the Context of Space Projects
- ❑ Block #2 – Software PA Organizational Aspects
- ❑ Block #3 – PA in the Software Life Cycle
- ❑ Block #4 – Software PA Across Processes

Software Product Assurance

Block 1 – Software in the Context of Space Projects

Software in the Context of Space Projects

- Introduction
- Software in the space project life cycle
- Space vs ground environment
- Software life-cycle & reviews

Importance of software in the system

Software implements (more and more of) the **system behaviour**

System **complexity** increases → software size increases

Software schedule is squeezed within the system schedule

Software is the last **flexibility** of the system at the end of the life cycle

Software is a candidate for **subcontracting** policies

Software touches many parts of the system. It has **interface** everywhere (ground – hardware – avionics – payloads – sensors – actuators – egse – security)

Software uses a **lot of data** from various system functional chains (centre of gravity, temperature, health status, voltage)

Software has several **users** (system – testing – operation)

What makes SW for aerospace special?

Embedded software

Challenging requirements

Scarce resources (CPU power, memory)

Harsh environment (radiation, thermal, vacuum …)

Late availability of the final platform

Difficult access to flight hardware

Schedule driven by a (rocket) launch date

Difficult access for SW maintenance (or no access at all)

SW is mission critical and/or impacting human safety

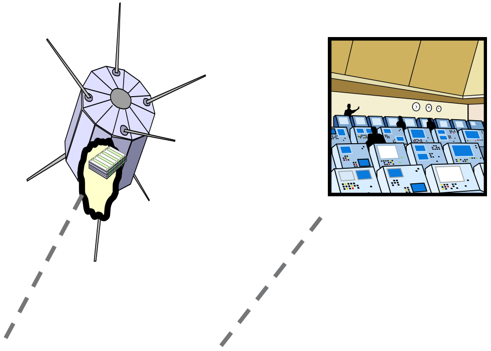
Unique development, reuse is difficult

Long development cycles (5 – 10 years)

Software and Space System Engineering

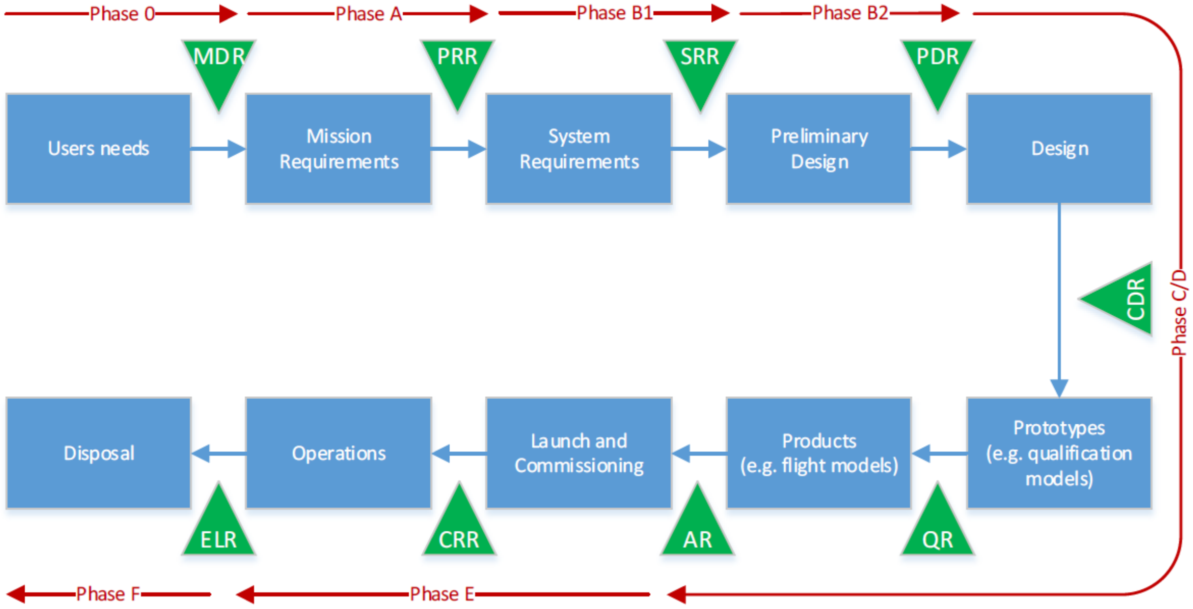
1. The software components of a space system play a role alongside the other engineering components such as mechanical and electrical

2. All of these various engineering components (including software) are governed by the overall discipline known as **space system engineering**



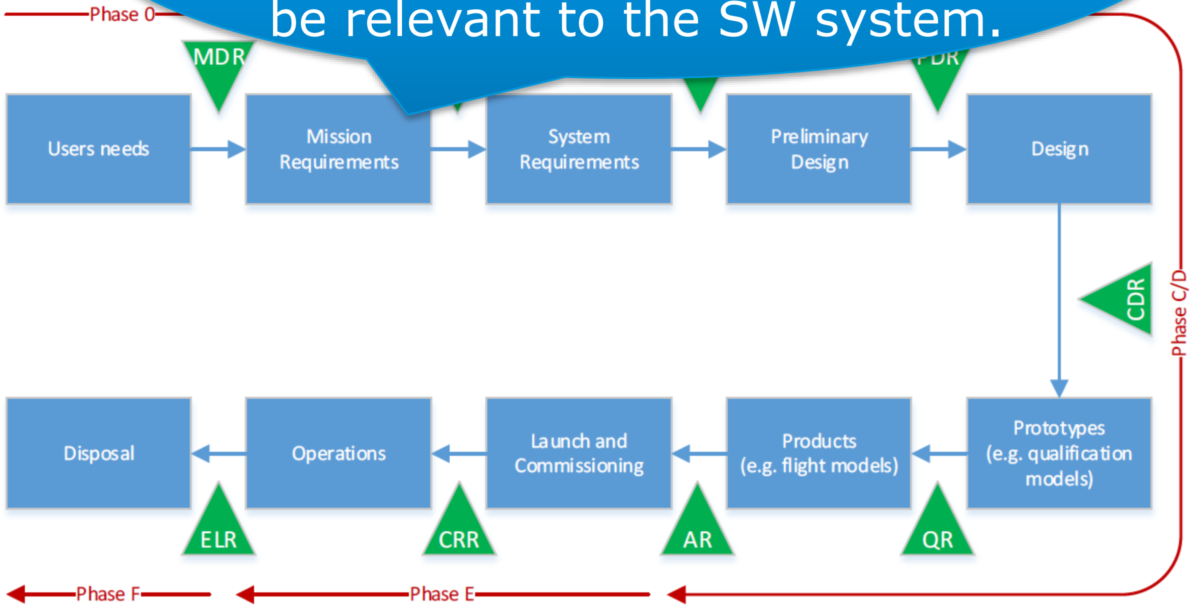
Software components are part of the overall mission system, together with other engineering components

SW in the space project life cycle



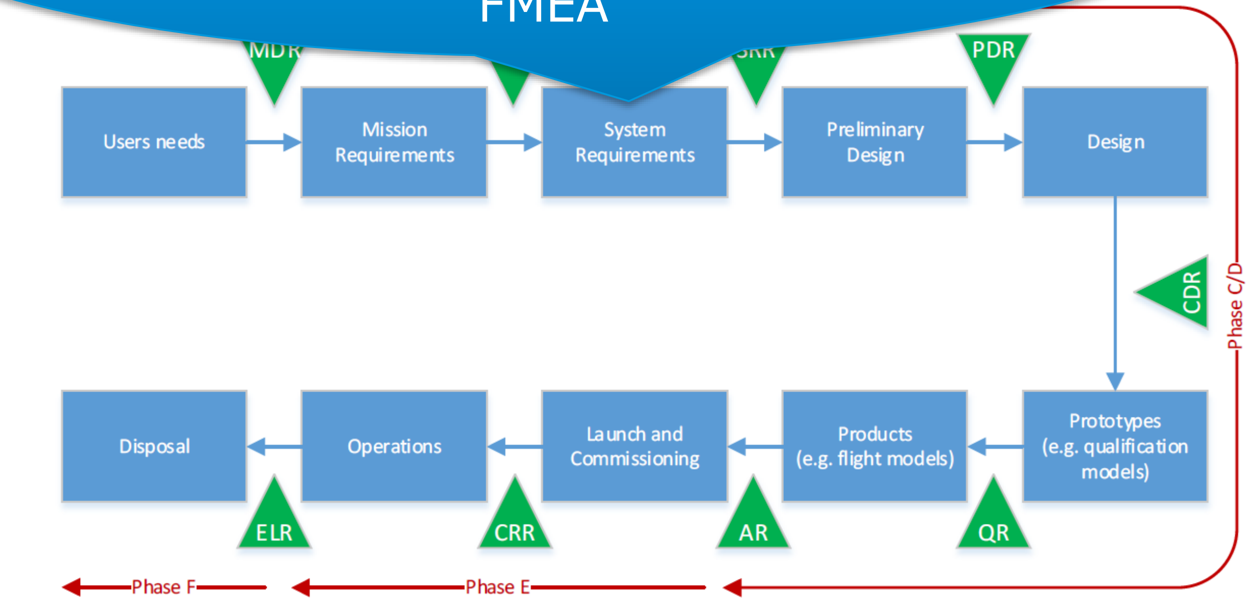
SW in the space

Budgets are defined (data rates, memory storage, processing power) which will be relevant to the SW system.



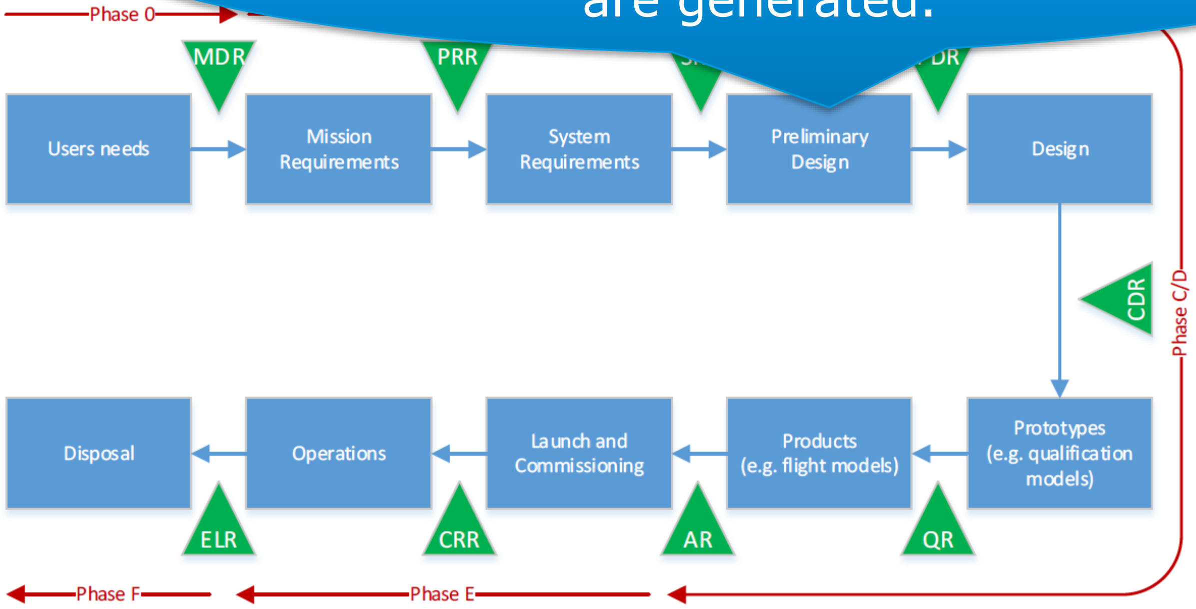
SW

Subsystems are identified, platform is fixed so related SW requirements are identified at high level. SW Criticality is determined based on the preliminary FMEA

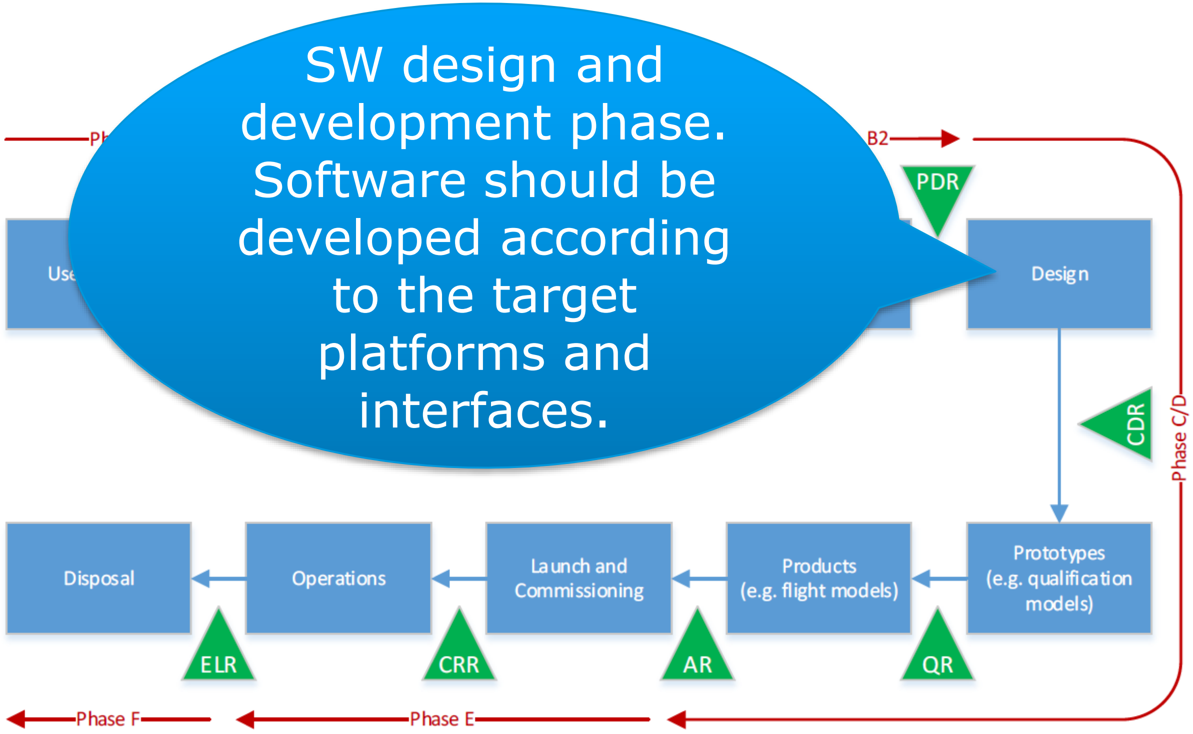


SW in the spa

SW system requirements are baselined. SW criticality refined from consolidated FMEA. Software technical specifications are generated.

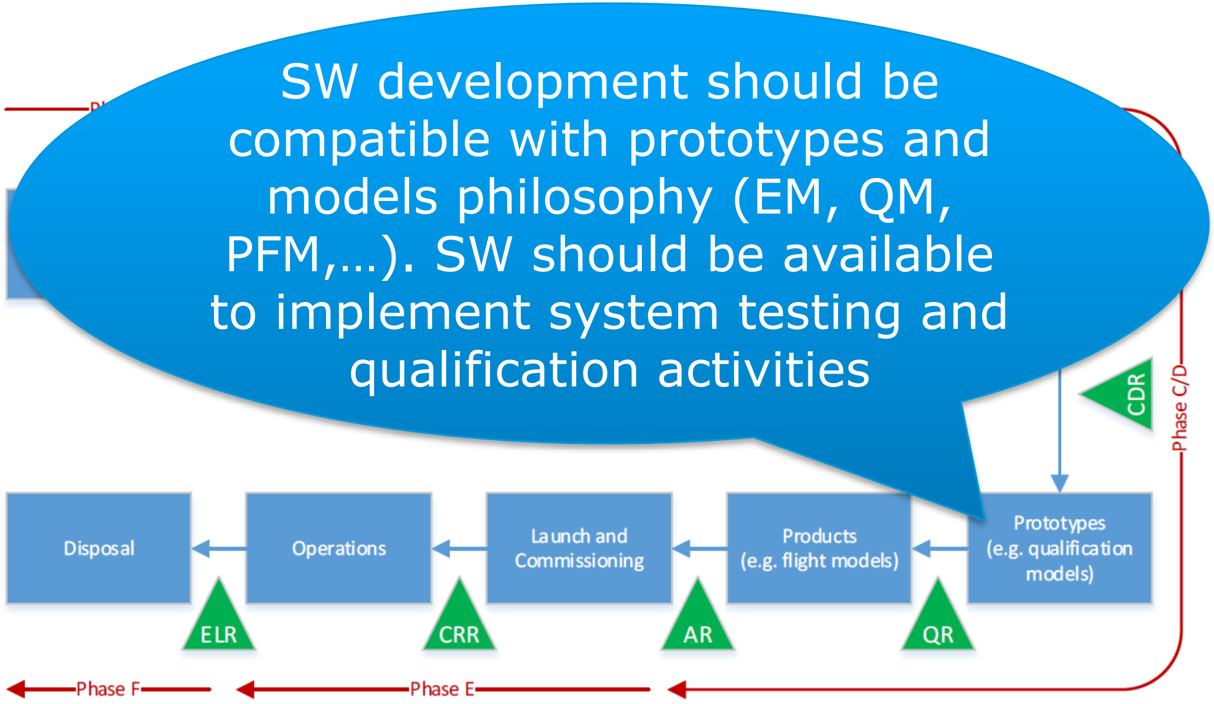


SW in the space project life cycle



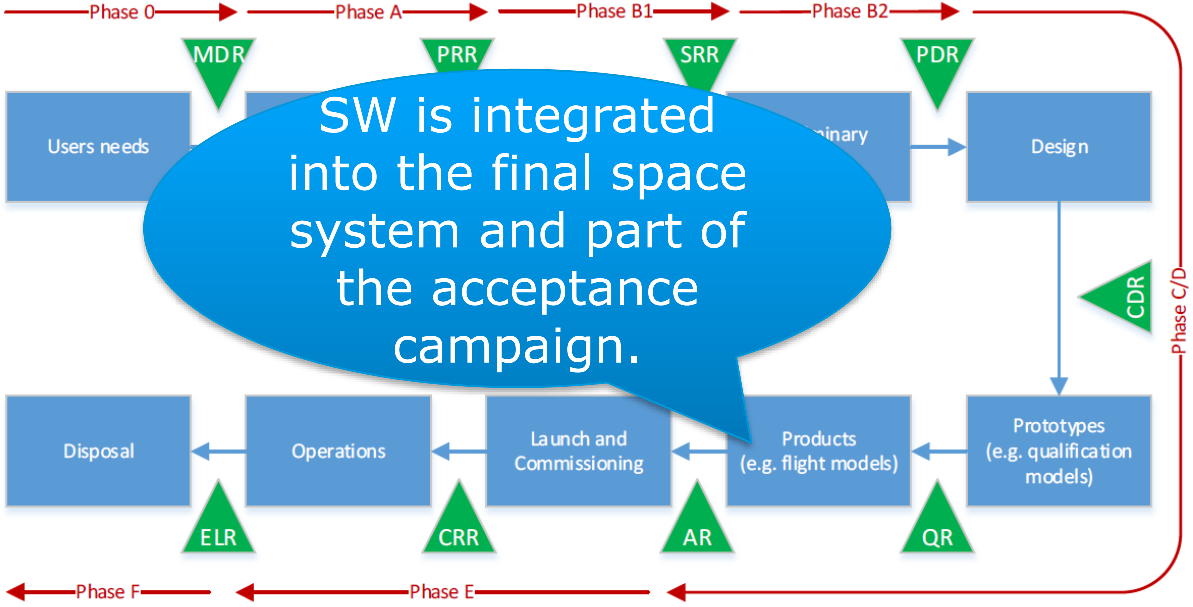
SW design and development phase. Software should be developed according to the target platforms and interfaces.

SW in the space project life cycle



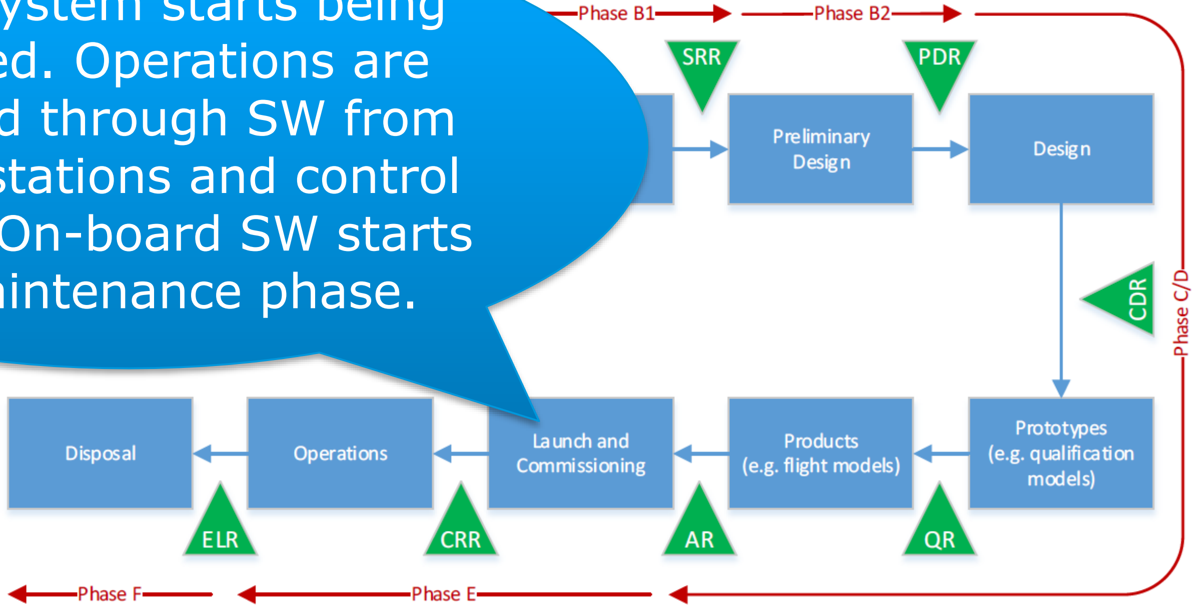
SW development should be compatible with prototypes and models philosophy (EM, QM, PFM,…). SW should be available to implement system testing and qualification activities

SW in the space project life cycle



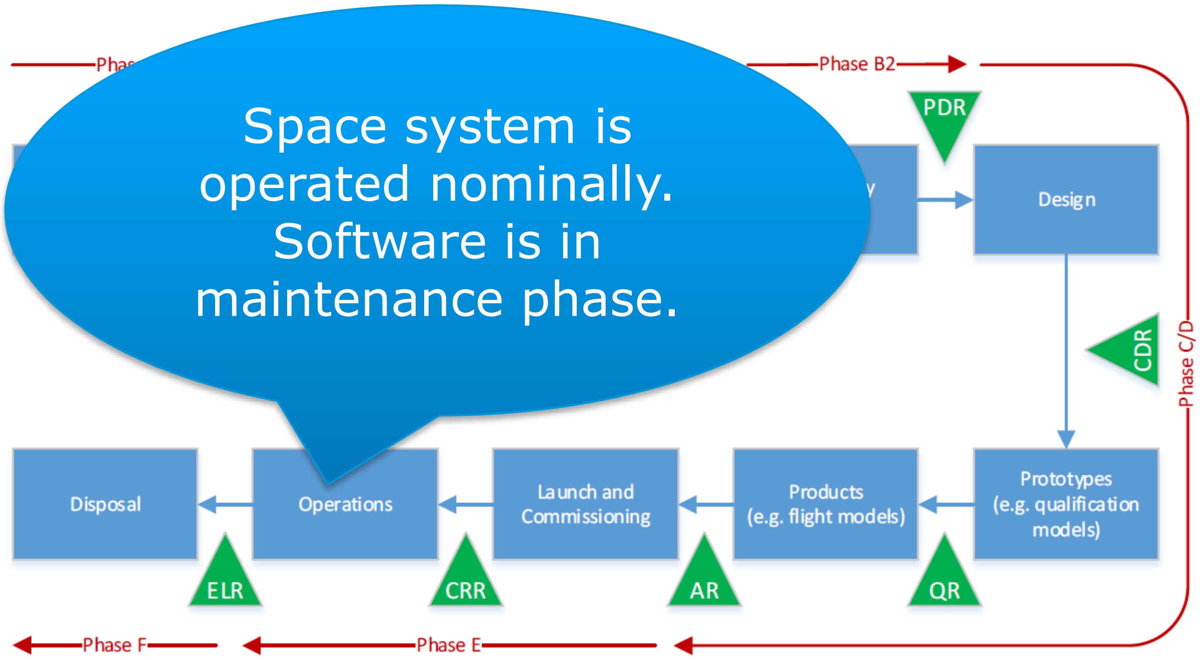
SW is integrated into the final space system and part of the acceptance campaign.

SW in the space project life cycle



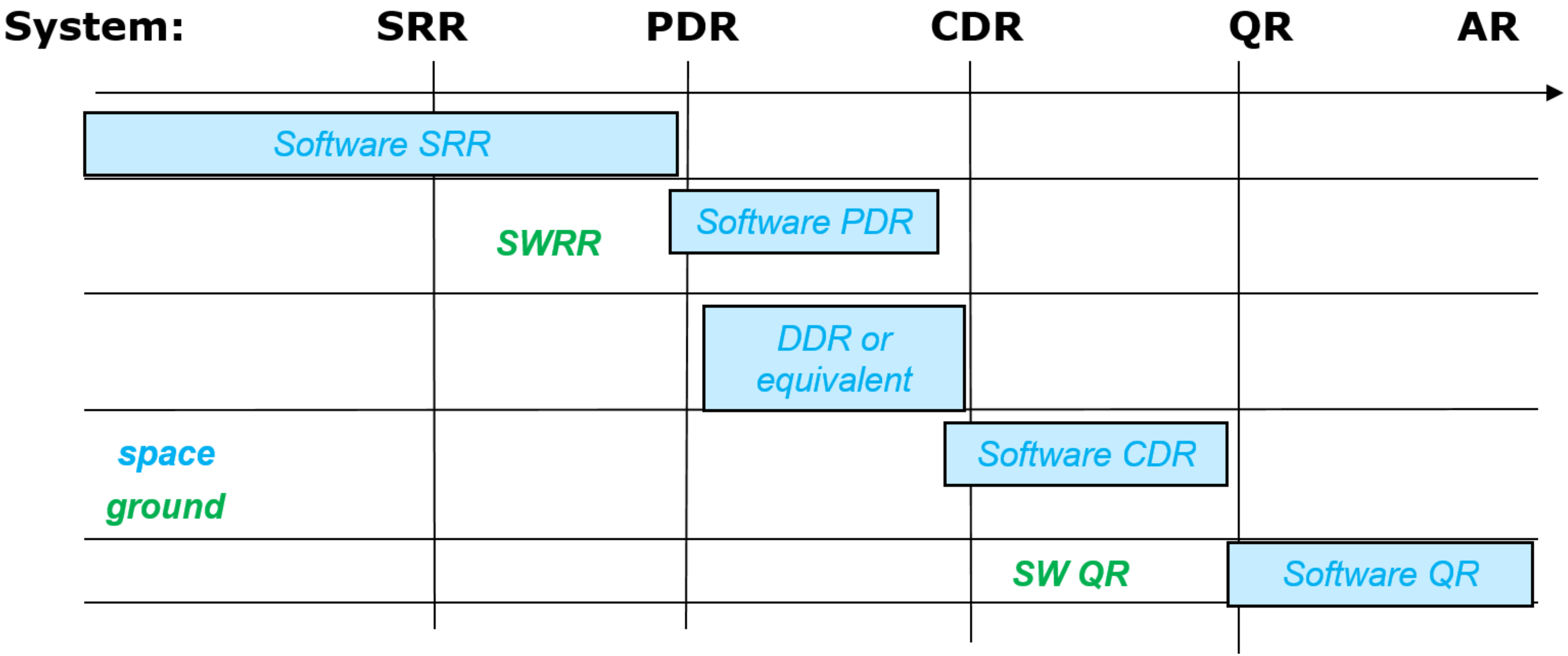
Space system starts being operated. Operations are managed through SW from ground stations and control centers. On-board SW starts the maintenance phase.

SW in the space project life cycle

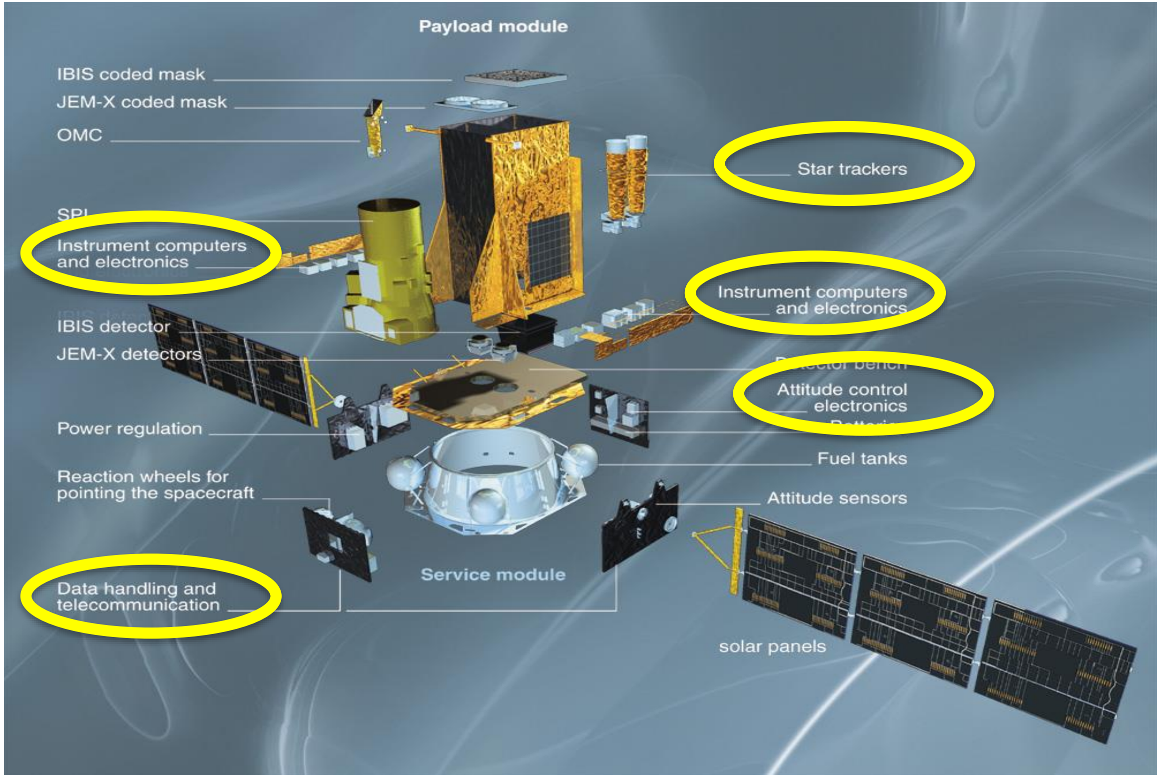


Space system is operated nominally. Software is in maintenance phase.

SW in the space project life cycle

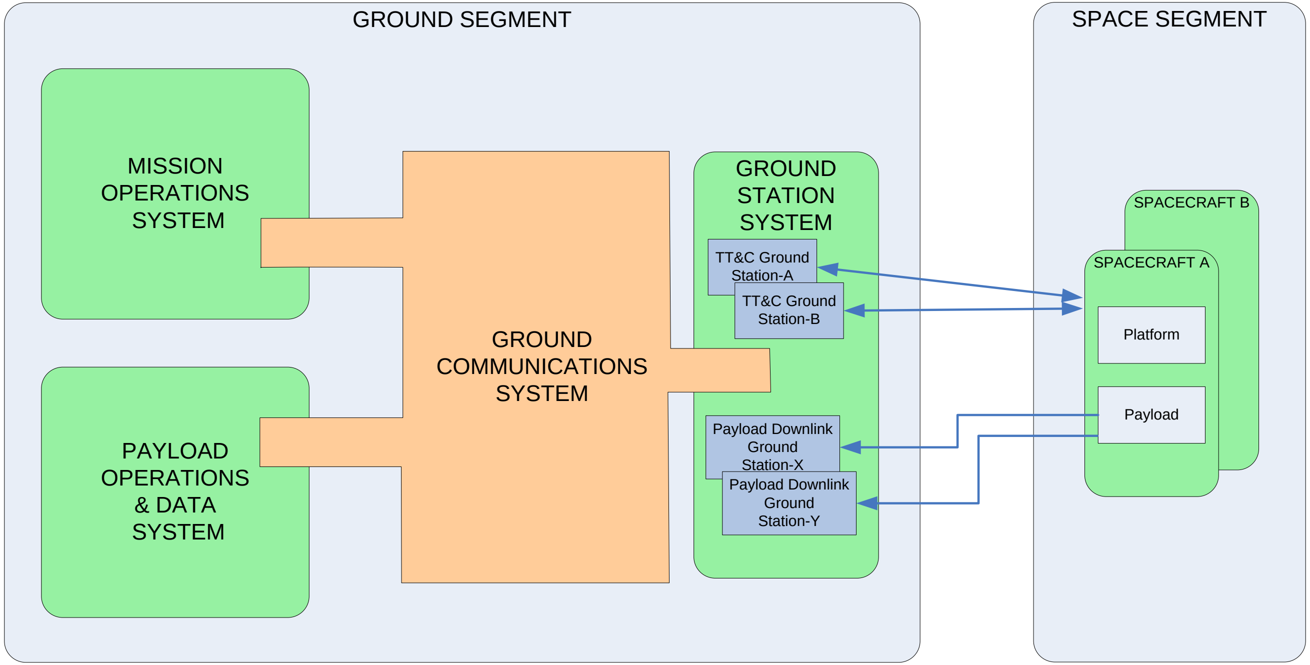


Space software environments (I)

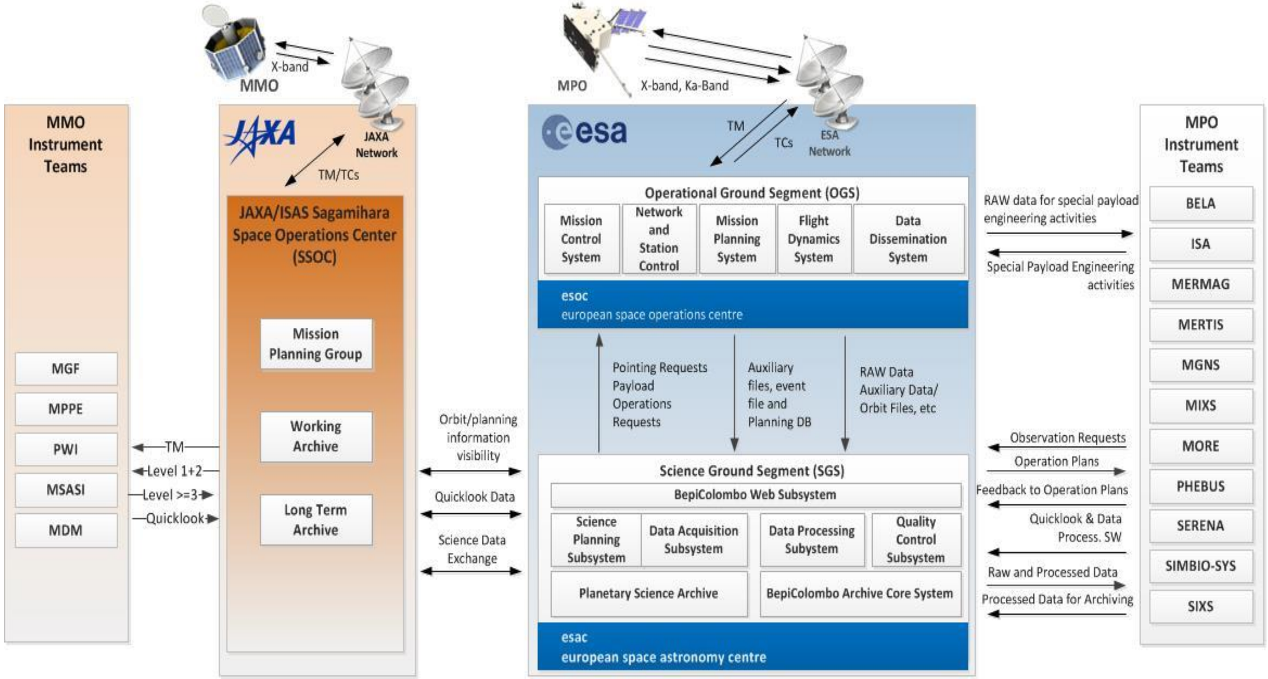


Space software environments (II)

Ground and space segment


GROUND SEGMENT
MISSION OPERATIONS SYSTEM
PAYLOAD OPERATIONS & DATA SYSTEM
GROUND COMMUNICATIONS SYSTEM
GROUND STATION SYSTEM
TT&C Ground Station-A
TT&C Ground Station-B
Payload Downlink Ground Station-X
Payload Downlink Ground Station-Y
SPACE SEGMENT
SPACECRAFT B
SPACECRAFT A
Platform
Payload

Bepi Colombo Ground Segment



MMO

X-band

MPO

X-band, Ka-Band

MMO Instrument Teams

- MGF
- MPPE
- PWI
- MSASI
- MDM

TM

Level 1+2

Level >=3

Quicklook

JAXA

JAXA Network

TM/TCs

JAXA/ISAS Sagamihara Space Operations Center (SSOC)

- Mission Planning Group
- Working Archive
- Long Term Archive

Orbit/planning information visibility

Quicklook Data

Science Data Exchange

esa

TM

TCs

ESA Network

Operational Ground Segment (OGS)

- Mission Control System
- Network and Station Control
- Mission Planning System
- Flight Dynamics System
- Data Dissemination System

esoc
european space operations centre

Pointing Requests Payload Operations Requests

Auxiliary files, event file and Planning DB

RAW Data Auxiliary Data/ Orbit Files, etc

Science Ground Segment (SGS)

- BepiColombo Web Subsystem
- Science Planning Subsystem
- Data Acquisition Subsystem
- Data Processing Subsystem
- Quality Control Subsystem
- Planetary Science Archive
- BepiColombo Archive Core System

esac
european space astronomy centre

RAW data for special payload engineering activities

Special Payload Engineering activities

Observation Requests

Operation Plans

Feedback to Operation Plans

Quicklook & Data Process. SW

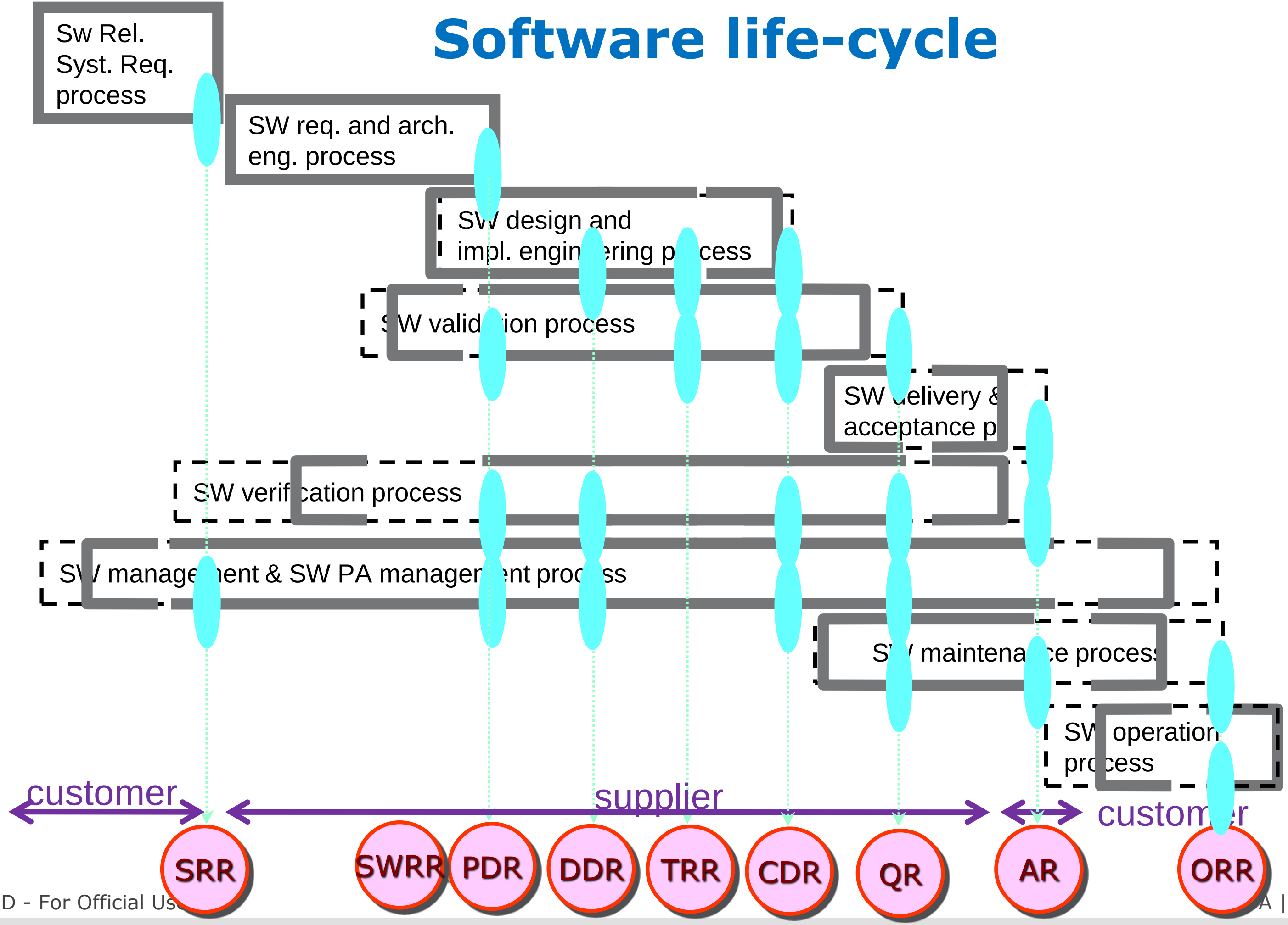
Raw and Processed Data

Processed Data for Archiving

MPO Instrument Teams

- BELA
- ISA
- MERMAG
- MERTIS
- MGNS
- MIXS
- MORE
- PHEBUS
- SERENA
- SIMBIO-SYS
- SIXS

Software life-cycle



Software project reviews

System requirement review (SRR)

After completion of the software requirements baseline specification, a system requirements review (SRR) shall take place.

<u>Typical objectives</u>:

- Agree with the customer or their representatives that all requirements captured in the requirements baseline are commonly understood and agreed.
- Review and baseline of the Requirements Baseline.
- Suitability of the draft software development plan including the software planning elements.
- Consistency of the software planning elements with respect to the upper level planning.
- Ensurance that software product assurance activities are performed.
- Evaluation of readiness to proceed to the next phase.

Software project reviews

Preliminary design review (PDR)

After completion of the software requirement analysis and architectural design, and the verification and validation processes implementation, a preliminary design review (PDR) shall take place.

Typical objectives (I):

- Agree with the customer or their representatives that all requirements with respect to the requirements baseline are captured in the technical specification.
- Review and baseline the Technical Specification.
- Review and baseline the software development approach and relevant plan.
- Review and baseline the software product assurance approach and relevant plan.
- Review and baseline the software configuration management approach and relevant plan.

Software project reviews

Preliminary design review (PDR) - cont

Typical objectives (II):

- Review and baseline the software verification and validation approach and relevant plan.
- Review and baseline of the software architecture.
- Review the technical budget and margins estimations.
- Review the integration strategy.
- Evaluation of the potential re-use of the software if applicable.
- Review of known unsolved issues which can have major impacts.
- Review the quality assurance reports.
- Evaluation of readiness to proceed to the next phase.

Software project reviews

Test readiness review (TRR)

A test readiness review (TRR) shall be held before the beginning of each significant test campaign, as defined in the software development plan.

Typical objectives:

- Baseline of the relevant test specifications (including test cases and procedures).
- Review of the testing facilities configuration.
- Verify that software documentation, software code, procured software and support software and facilities are under proper configuration control.
- Baseline the testing configuration.
- Review the quality assurance reports.
- Review the status of all open problems (SPRs, NCRs).
- Evaluation of readiness to proceed to testing.

Software project reviews

Critical design review (CDR)

After completion of the design of software items, coding and testing, integration and validation with respect to the technical specification, a critical design review (CDR) shall take place.

Typical objectives (I):

- Baseline of the detailed design (including the verification reports and technical budget report).
- Adequacy of the software units and integration plans and of the included unit and integration test procedures.
- Review and baseline the SValP approach and relevant plan.
- Review of the Software Reuse File, evaluation of the potential re-use of the software intended for reuse.
- Baseline of the validation specification w.r.t. the technical specification.
- Review of the unit and integration test results, including as-run procedures.

Software project reviews

Critical design review (CDR) - cont

Typical objectives (II):

- Verification that all the Technical Specification has been successfully validated (validation report) and verified (including technical budget, memory and CPU, and code coverage).
- Verify that the Software Configuration Item under review is a formal version under configuration control.
- Review of the software user manual.
- Review of known unresolved issues which can have major impact and resolution plan identification.
- Review the quality assurance reports.
- Review of the RB-validation facilities.
- Baseline of the Validation specification against the RB.
- Evaluation of readiness to proceed to the next phase.

Software project reviews

Qualification review (QR)

After completion of the software validation against the requirements baseline, and the corresponding verification activities, a qualification review (QR) shall take place.

Typical objectives (I):

- To verify that the software meets all of its specified requirements, and in particular that verification and validation process outputs enable transition to ”qualified state” for the software products.
- Review of the RB-validation test, analysis, inspection or review of design results, including as-run procedures.
- Verification that all the Requirements Baseline and interfaces requirements have been successfully validated and verified (including technical budgets and code coverage).

Software project reviews

Qualification review (QR) - cont

Typical objectives (II):

- Review the software release document.
- Review of the acceptance facilities configuration.
- Verify that the Software Configuration Item under review is a formal version under configuration control.
- Review of known unresolved issues which can have major impact and resolution plan identification.
- Review the quality assurance reports.
- Evaluation of readiness to proceed to the next phase.
- Review of the maintenance plan.
- Review the acceptance test plan.

Software project reviews

Acceptance review (AR)

After completion of the software delivery and installation, and software acceptance, an acceptance review (AR) shall take place.

Typical objectives (I):

- Review of the acceptance test results, including as-run procedures.
- Verify that the Software Configuration Item under review is a formal version under configuration control.
- Verification that all the RB software requirements have been successfully validated and verified (including technical budgets and code coverage) throughout the development life cycle.
- Baseline of the software acceptance data package.

Software project reviews

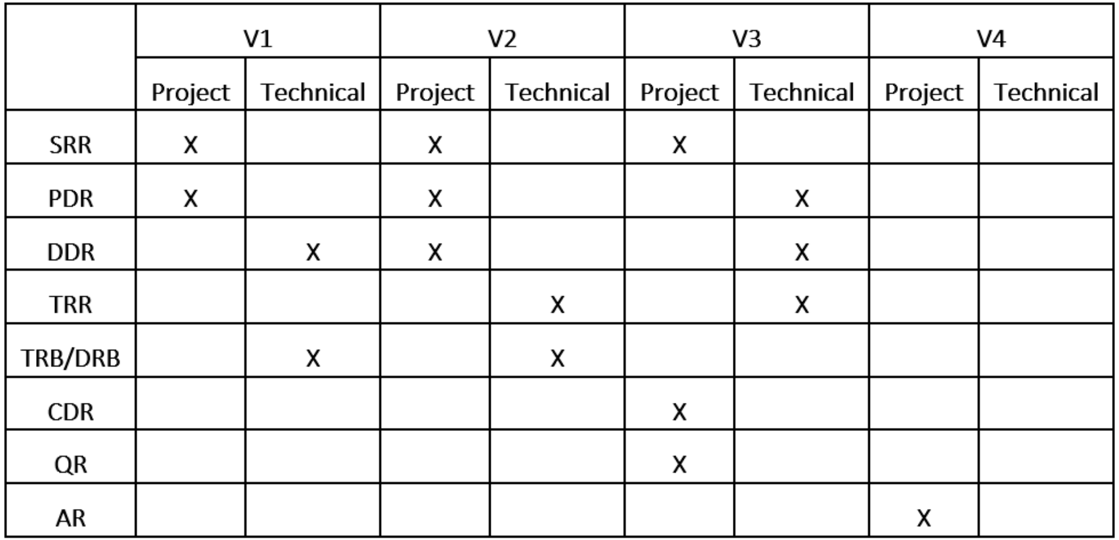
Acceptance review (AR) - cont

Typical objectives (II):

- Verify that the complete set of acceptance test cases is run on the same software version.
- Acceptance of the software product.
- Review of the software release document, the installation procedure and report and the maintenance plan.
- Review of known unresolved issues which can have major impact and identification of resolution plan for each outstanding issue and known problems.
- Correct closure of major SPRs/NCRs.
- Review of RFWs.
- Review the quality assurance reports.

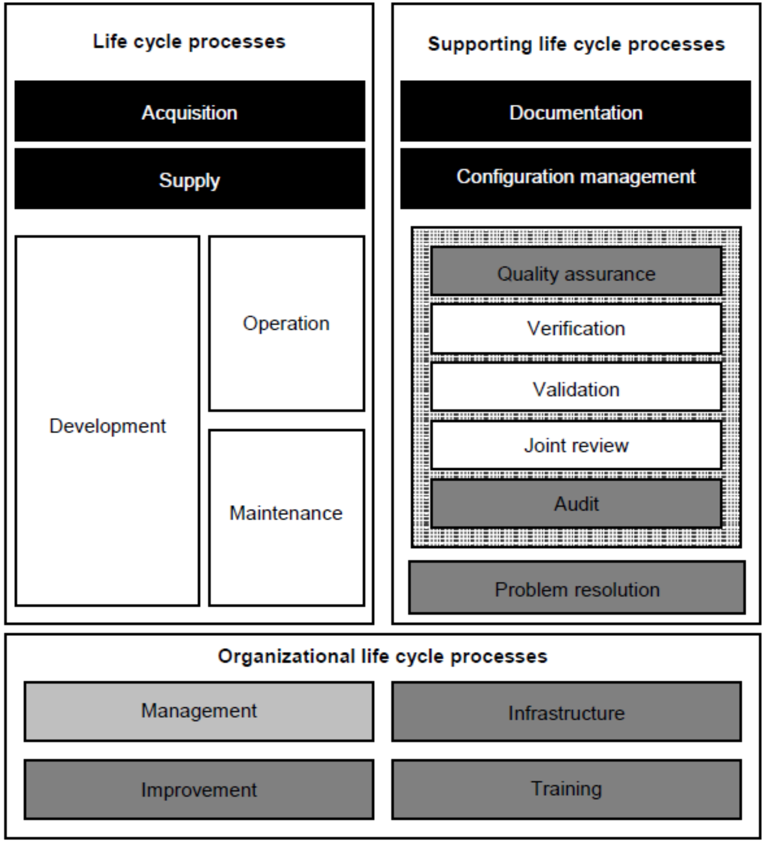
Software life-cycle models

- Life-cycle models discussed in software engineering handbook ECSS-E-HB-40A.
- Additional guidance on ISO/IEC TR 24748-3:2011. Guide for ISO/IEC 12207 (Software Life Cycle Processes).
- Project vs technical review setup example:



	V1		V2		V3		V4	
	Project	Technical	Project	Technical	Project	Technical	Project	Technical
SRR	X		X		X			
PDR	X		X			X		
DDR		X	X			X		
TRR				X		X		
TRB/DRB		X		X				
CDR					X			
QR					X			
AR							X	

Mapping of ECSS processes to ISO/IEC 12207



Life cycle processes

Acquisition

Supply

Development

Operation

Maintenance

Supporting life cycle processes

Documentation

Configuration management

Quality assurance

Verification

Validation

Joint review

Audit

Problem resolution

Organizational life cycle processes

Management

Infrastructure

Improvement

Training

■ Other ECSS

□ ECSS-E-ST-40

▩ ECSS-Q-ST-80

▨ Details for SPA and/or SWE

Software processes in ECSS-E-ST-40C

5.2 Software related system requirements

- 5.2.2 Sw. rel. Syst. req. analysis
- 5.2.3 Sw. rel. system verification
- 5.2.4 Sw. rel. system integration & ctrl
- 5.2.5 System Requirement Review

5.4 SW req. & arch. engineering process

- 5.4.2 Software requirements analysis
- 5.4.3 Software architectural design
- 5.4.4 Preliminary Design Review

5.5 SW des. & impl. engineering process

- 5.5.2 Design of software items
- 5.5.3 Coding and testing
- 5.5.4 Integration

5.6 Software validation process

- 5.6.2 Validation process implementation
- 5.6.3 Validation w.r.t. the technical spec.
- 5.6.4 Validation w.r.t. the req. baseline

5.7 Software delivery and acceptance process

- 5.7.2 Software delivery and installation
- 5.7.3 Software acceptance

5.8 Software verification process

- 5.8.2 Verification process implementation
- 5.8.3 Verification activities

5.9 Software operations process

- 5.9.2 Process implementation
- 5.9.3 Operational testing
- 5.9.4 Software operation support
- 5.9.5 User support

5.10 Software maintenance process

- 5.10.2 Process implementation
- 5.10.3 Problem & modific. analysis
- 5.10.4 Modification implementation
- 5.10.5 Conducting mainten. reviews
- 5.10.6 Software migration
- 5.10.7 Software retirement

5.3 Software management process

- 5.3.2 Sw life cycle managmt.
- 5.3.3 Joint review process
- 5.3.4 Sw. Proj. Rev. Descr.
- 5.3.5 Sw Tech. Rev. Descr.
- 5.3.6 Review Phasing
- 5.3.7 Interface management
- 5.3.8 Tech. bdg & margin mngt
- 5.3.9 Compliance to Standard

Relevant ECSS standards and handbooks (ecss.nl)

ECSS-E-ST-40C - Software

ECSS-Q-ST-80C Rev. 1– Software product assurance

ECSS-E-ST-10C - System engineering general requirements

ECSS-M-ST-40C - Configuration and information management

ECSS-Q-HB-40A – Software engineering handbook

ECSS-Q-HB-40-01A - Agile software development handbook

ECSS-Q-HB-80-01A – Reuse of existing software

ECSS-Q-HB-80-01 1A/2A - Software process assessment and improvement

ECSS-Q-HB-80-03A Rev.1 - Software dependability and safety

ECSS-Q-HB-80-04A - Software metrication programme definition and implementation

Disclaimer

This presentation is a property of the European Space Agency (ESA) or ESA's licensors. No part of this material may be reproduced, displayed, amended, distributed or otherwise used in any form or by any means, without written permission of ESA or ESA's licensors. Any unauthorised activity or use shall be an infringement of ESA's or ESA licensors' intellectual property rights and ESA reserves the right to defend its rights and interests, including to seek for remedies.