An Overview of ESA Software Product Assurance Services

M. Garcia, R. Prades
ESA D/TEC-QQS

SAS 2007
Morgantown, West Virginia, USA
September 25-27, 2007

Presentation Overview

- **Software product assurance context in ESA**
- **Software product assurance services provided to European space projects and industry**
  - Software Product Evaluation for Conformity (SPEC)
  - Software Safety and Dependability Evaluation (RAMS)
  - Software Process Assessment (S4S)

SW PA Context in ESA

- **ESA Directorate Organization**
  - Program Directorates: D/SCI, D/EOP, D/HME, …
  - Technical Directorate: D/TEC
    - Systems design
    - Electrical Engineering
    - Mechanical Engineering
    - Quality
      - Materials
      - Component
      - Software, Dependability and Safety
- **Project teams/technical support**

SW PA Context in ESA

- **D/TEC-QQS**
  - SW PA support to projects
    - Meeting requirements specified in applicable documentation → development of methods, approaches, tools
  - R&D in the SW PA domain with focus on present and future needs of projects and industrial partners
    - Handbooks containing guidance on how to apply SW PA in projects
    - Services
  - Services
    - Software Product Evaluation for Conformity
    - Software Safety and Dependability Evaluation
    - Software Process Assessment

Software Process Assessment (S4S)

Software Process Assessment (S4S)

- **What is S4S?**
  - A method of evaluating space software processes
  - ISO 15504 (Process Assessment) conformant
  - Can be tailored to different classes of safety-critical software
- **Scope**
  - Processes include acquisition, supply, operation, engineering, supporting, management, process improvement, resource and infrastructure and reuse
  - Can select which processes to assess
  - Each process is assessed on a scale of capability

Software Process Assessment (S4S)

- **25 assessment performed**
  - Prime contractors
  - Small and medium enterprises
- **Performed on a voluntary basis**
  - few in response to specific project requirements
  - Input to improvement programs
  - Confirmation of SW development capabilities towards customers

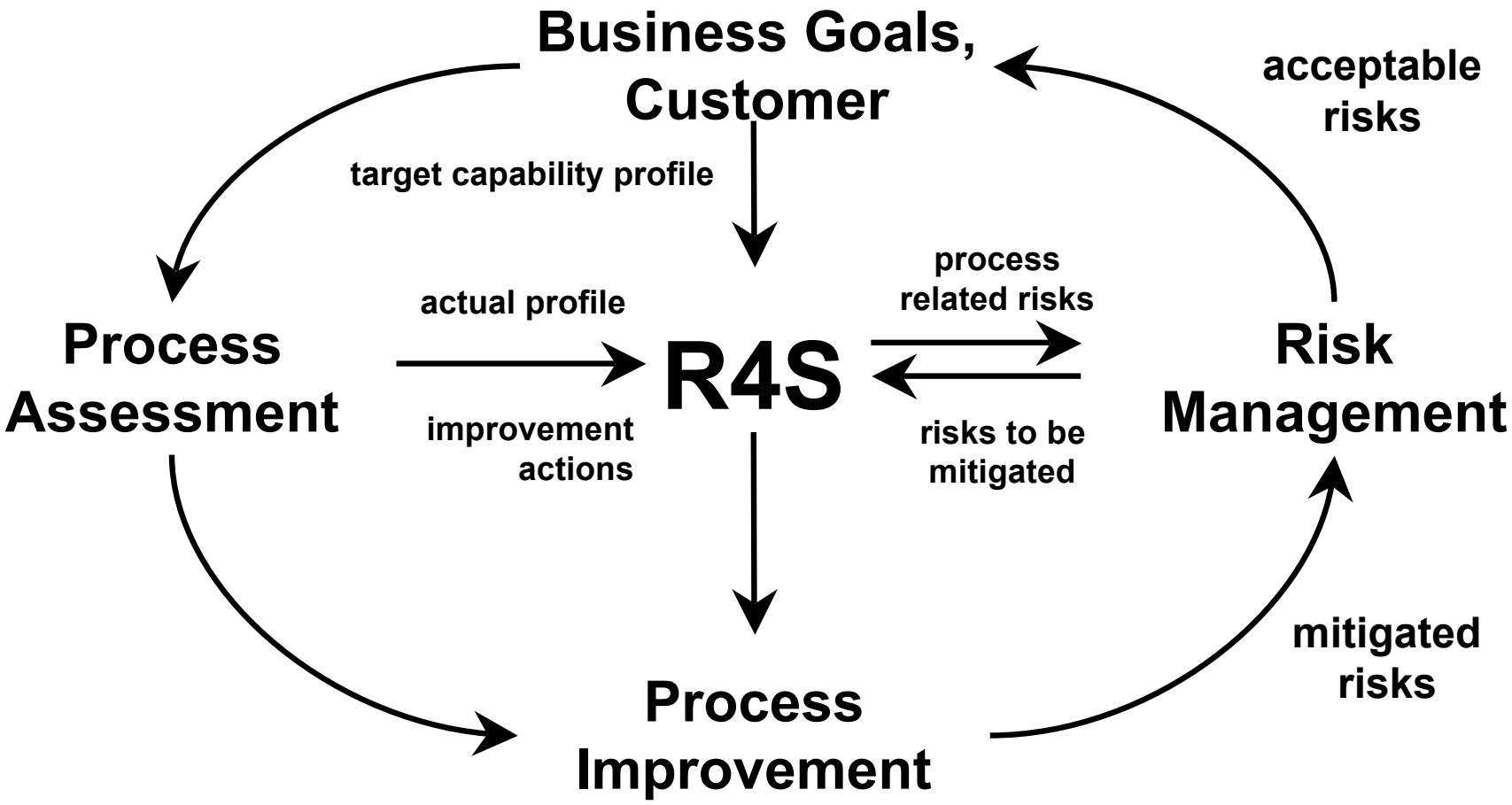
Software Process Assessment (S4S)

- **Assessment organization**
  - Pre-assessment visit
    - Selection of processes
    - Determination of the capability levels to assess
    - Schedule and participants
  - Assessment
- **Improvement Workshop**
  - Identification of improvement opportunities from assessment report
  - Discussion and setup of improvement projects

Risk Reduction with S4S

- **What is R4S?**
  - An extension of S4S with a risk dimension integrated in the method
  - Basic idea: use risk analysis post-assessment to identify most critical processes for improvement
  - Relies on correlations between commonly known risks & S4S processes
- **Objectives**
  - Identification, assessment and management of process related risks
  - Link between process and risk management
  - Supports identification and prioritization of improvement actions to
    - improve processes towards target capability levels
    - reduce process-related risks to an acceptable level

Risk Reduction with S4S



Software Product Evaluation for Conformity (SPEC)

Software Product Evaluation for Conformity (SPEC)

- **ESA's Initiative on Software Product Evaluation**
  - Provide a framework to evaluate, and possibly confirm conformity of a space software product.
  - Provide a good bases for specifying the non-functional requirements through a quality model
  - Reduce significantly the cost associated with the product development by adopting a quality model from the beginning of a project.
  - Increase the ability to produce quality software and to specify and assess this quality

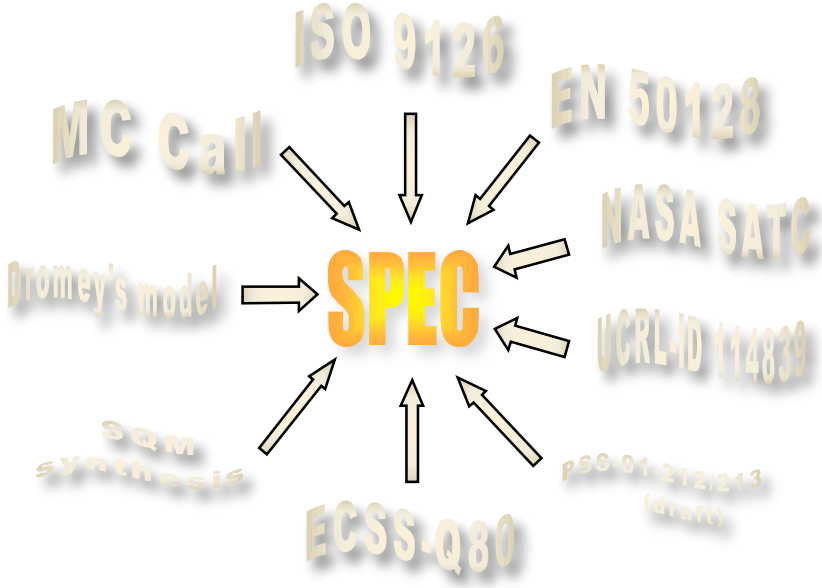
Software Product Evaluation for Conformity (SPEC)
The context

- **ECSS-Q80B "Software product assurance," defines requirements on Software product quality assurance**
  - Definition or adoption of a quality model
  - Definition of a metrication program
  - Definition of metrics
  - Verification of the achievements
  - Metrics trend
  - Improvement actions
- **HB-Q80-04 "Software metrication programme definition and implementation"**

Software Product Evaluation for Conformity (SPEC)
SPEC method

- **SPEC is a method to evaluate a Space Software Product by measuring its quality, based on a tailored quality model.**

- **SPEC defines:**
  - Quality Model
  - Evaluation framework



Software Product Evaluation for Conformity (SPEC)
SPEC on on-board OS

- **Goals of the evaluation:**
  - Check the usability of the SPEC scheme for Open Source Software
  - Check whether the OS satisfies the SPEC criteria for mission critical software.
  - Obtain substantial information about its features and determine potential areas for (future) in depth analysis
  - Identify weaknesses and potential areas of improvements for the SPEC scheme

Software Product Evaluation for Conformity (SPEC)
SPEC on on-board OS

- **Constraints of the evaluation**
  - Product was an Open Source Software with a limited set of documentation
  - The provider did not issue any documentation about the verification activities (methodology and results)
- **Evaluation results**
  - Methodological SPEC related results (SPEC to OSS)
  - Software product (On-board OS) related results

Software Product Evaluation for Conformity (SPEC)
SPEC on GSI

- **Goals of the evaluation**
  - Evaluation of GSI in a generic scenario (i.e. typical scientific mission)
  - Application of SPEC to real project
  - SPEC as a mean to improve software products
  - Use of S4S* for process related goal properties

(*) Spice for Space : Process assessment method compliant with ISO15504

Software Product Evaluation for Conformity (SPEC)
SPEC on SGI -- Phases

Phase I: Baseline Evaluation

- Tailoring of Quality Model (based on criticality classes)
- Evaluation Plan & Data Collection
- Evaluation Report (with recommendations)

Phase II: Improvement

- Implementation of recommendations (including specific process improvement actions)

Phase III: Delta Evaluation

- Re-measurement phase (including delta S4S)
- Re-issue of Evaluation Report (21 detailed recommendations)

Software Product Evaluation for Conformity (SPEC) Conclusions

- Identification of recommendations allows the implementation of improvements in some processes of the software development and in further versions of the software products.
- Although SPEC was not meant to be used “after the fact” on existing products, it was demonstrated that SPEC can be applied successfully on finalized SW products.
- SPEC needs to be tailored before it can be used to evaluate OSS.
- SPEC is being updated based on the experience gained during its application on real projects.

Software Safety and Dependability Evaluation

Software Safety and Dependability Evaluation

- **ESA's Initiative on Software Dependability and Safety Evaluations**
  - Evaluate dependability and safety of software to reduce the risk of mission failures
  - Help assure correct implementation of system PA requirements
  - Evaluate the applicability and usefulness of selected techniques
  - Identify improvements of the techniques

Software Safety and Dependability Evaluation
The context

- **ECSS-Q80B "Software product assurance," defines requirements on safety and mission critical software**
  - Identification of the criticality of the software function in the system context
  - Software dependability and safety analysis to be performed
  - Design of the software to minimize number of critical components
  - Use of criticality for tailoring of the "development rigour"
  - Special rules for handling of critical software components
  - Organizational constraints
- **HB-Q80-03 " Methods and techniques to support the assessment of software dependability and safety"**

Software Safety and Dependability Evaluation
SFMECA on GSI

- Software Failure Modes, Effects and Criticality Analysis performed on the Spacecraft Control and Operations System used for spacecraft operation
  - A typical scientific mission was used as a reference
  - Performed starting from top-level requirements, then analyzing the failure modes of the SW components
- Main result was the criticality level assigned to SW components based on the severity of the consequences of failures

Software Safety and Dependability Evaluation
Code Analysis on GSI

- Applied to components having highest criticality and highest density of severe failure modes associated
- Based on checklist derived from applicable coding standards
- Main output was the report on coding standard rules violated and metrics values.

Software Safety and Dependability Evaluation
Robustness and stress testing of on-board OS

- This off-the-shelf open-source real time operating system is broadly used in space applications
- Tested against the available “specifications”
- Robustness-tested with singular and boundary input values, exercising error handling mechanisms
- Stress-tested in conditions of extreme workload
- Faults mainly from robustness tests:
  - Incorrect control flow
  - Memory alignment/Illegal instruction exceptions
  - Unexpected error codes returned

Software Safety and Dependability Evaluation
HSIA on scientific instrument

- Hardware-Software Interaction Analysis performed on a science payload of the space observatory
- HW failure modes extracted from FMECA (some newly identified from system analysis)
- Several issues identified:
  - Lack of failure detection and recovery mechanisms
  - Deficient failure reporting
  - System FMECA Report updates necessary

Software Safety and Dependability Evaluation
Hazard Analysis on a Flight Application Software

- Methodology used from ESA standard HB-Q80-03
- Based on list of feared events and severity levels derived from system-level analyses
- Software Fault Tree Analysis starting from selected feared event, to identify basic events; then SW FMECA on the SW functions/components that can be traced to the basic events
- As a result, the combination of techniques allowed the identification of potential hazard scenarios and investigation of hazard reduction/control mechanisms

Software Safety and Dependability Evaluation Conclusions

- Systematic identification of SW failure modes and criticality classification of SW components
- To drive the verification activities and the “rigour” of the development, based on the criticality classification of the SW components
- Verification that for all potential HW failures the SW reaction is correctly specified
- Improvement of system-level Failure Detection, Isolation and Recovery (FDIR) through Integration of SW-level analyses results into system-level analyses