Certifications of Critical Systems – The CECRIS Experience

RIVER PUBLISHERS SERIES IN INFORMATION SCIENCE AND TECHNOLOGY

Series Editors

K. C. CHEN
National Taiwan University
Taipei, Taiwan

SANDEEP SHUKLA
Virginia Tech, USA
and
Indian Institute of Technology Kanpur, India

Indexing: All books published in this series are submitted to Thomson Reuters Book Citation Index (BkCI), CrossRef and to Google Scholar.

The "River Publishers Series in Information Science and Technology" covers research which ushers the 21st Century into an Internet and multimedia era. Multimedia means the theory and application of filtering, coding, estimating, analyzing, detecting and recognizing, synthesizing, classifying, recording, and reproducing signals by digital and/or analog devices or techniques, while the scope of "signal" includes audio, video, speech, image, musical, multimedia, data/content, geophysical, sonar/radar, bio/medical, sensation, etc. Networking suggests transportation of such multimedia contents among nodes in communication and/or computer networks, to facilitate the ultimate Internet.

Theory, technologies, protocols and standards, applications/services, practice and implementation of wired/wireless networking are all within the scope of this series. Based on network and communication science, we further extend the scope for 21st Century life through the knowledge in robotics, machine learning, embedded systems, cognitive science, pattern recognition, quantum/biological/molecular computation and information processing, biology, ecology, social science and economics, user behaviors and interface, and applications to health and society advance.

Books published in the series include research monographs, edited volumes, handbooks and textbooks. The books provide professionals, researchers, educators, and advanced students in the field with an invaluable insight into the latest research and developments.

Topics covered in the series include, but are by no means restricted to the following:

- Communication/Computer Networking Technologies and Applications
- Queuing Theory
- Optimization
- Operation Research
- Stochastic Processes
- Information Theory
- Multimedia/Speech/Video Processing
- Computation and Information Processing
- Machine Intelligence
- Cognitive Science and Brian Science
- Embedded Systems
- Computer Architectures
- Reconfigurable Computing
- Cyber Security

For a list of other books in this series, www.riverpublishers.com

Certifications of Critical Systems – The CECRIS Experience

Editors

Andrea Bondavalli

Consorzio Interuniversitario Nazionale per l'Informatica (CINI)
and University of Florence
Italy

Francesco Brancati

ResilTech Srl
Italy

Published, sold and distributed by:
River Publishers
Alsbjergvej 10
9260 Gistrup
Denmark

River Publishers
Lange Geer 44
2611 PW Delft
The Netherlands

Tel.: +45369953197
www.riverpublishers.com

ISBN: 978-87-93519-56-5 (Hardback)
978-87-93519-55-8 (Ebook)

Printed on acid-free paper.

Contents

Preface

The rapid spread of critical systems raises new challenges from multiple aspects. The functionality embedded into critical systems is a major driver of efficient and economic operation of a variety of societal services ranging from traffic control to health care, but at the same time, the vulnerability of the society to malfunctioning equipment reaches a critical level both in the terms of risks to the human life and huge economic impacts. The rapid development of underlying technologies implies a huge challenge to this industry which followed for decades a safety driven conservative approach. This way, a uniform approach to the development, validation and verification is an important factor in the Europe wide integration of services as emphasized for instance by the creation of the ARTEMIS European Technology Platform on the side of technology. On the human skill side, the dissemination of the best industrial practices and appropriate training is a key enabling factor for this unification process.

All over Europe there is a significant lack of skilled workforce related to critical embedded systems.

Traditional V&V methods frequently exceed effort needed for the core development time, and while the "soft" IT industry rapidly turns to system integration based on the reuse of high volume hardware and software components, for safety related applications this will still evolve.

All this poses serious difficulties to companies, which are on one hand constrained to meet predefined quality goals, whereas, on the other hand, are required to deliver systems at acceptable cost and time to market. Large companies mainly follow a brute-force approach by focused large volume investment into tooling and in-house training, but even high-tech SMEs are highly vulnerable to the new challenges.

Looking at the field of the Verification and Validation one of the most challenging goals is the definition of methods, strategies and tools able to validate a system adequately, while simultaneously keeping the cost and delivery time reasonably low. It is not easily possible to establish a proper balance between achievable quality with a particular technique (in terms of RAMS

attributes) and the costs required for achieving such quality. The situation is even worse in the case of integration of existing SW in a safety critical system to be certified, since, assessing products which encompass COTS software is a challenge although modern standards consider this possibility. An additional concern is the usage of recently adopted methods for SW development like model based ones, since the certification of systems using software developed with these supports is at the limit of the applicability of the existing standards, and only the most recent ones are aligned with these 'modern' methods.

This book documents the main insights on Cost Effective Verification and Validation processes that we gained during our work in the European Research Project CECRIS (acronym for *Certification of Critical Systems*). The objective of this research was to tackle the challenges of certification by focusing on those aspects that turn out to be more difficult and or important for current and future critical systems industry: the effective use of methodologies, processes and tools.

The CECRIS project took a step forward in the growing field of development, verification and validation and certification of critical systems. It focused on the more difficult/important points of (safety, efficiency, business) of critical system development, verification and validation and certification process. The scientific objectives of the project were to study both the scientific and industrial state of the art methodologies for system development and the impact of their usage on the verification and validation and certification of critical systems. Moreover the project aimed at developing strategies and techniques supported by automatic or semi-automatic tools and methods for these types of activities, whose cost-quality achievements are well-predictable in order to tie costs of application of techniques to the RAMS attributes level achieved by the product being tested. The project set guidelines to support engineers during the planning of the verification & validation phases.

The Project Consortium was composed by three academic partners and three companies:

1. CINI-Consorzio Interuniversitario Nazionale per l'Informatica
2. Resiltech S.r.l.
3. Universidade de Coimbra
4. Budapesti Muszaki es Gazdasagtudomanyi Egyetem
5. Prolan Iranyitastechnikai Zartkoruen Mukodo Reszvenytarsasag
6. CRITICAL Software SA

The CECRIS project has given to the partners the opportunity of sharing their industrial-academic expertise and experiences and to develop fruitful collaborations and research products. Through the 'Transfer of Knowledge' activities, industrial partners have had the opportunity to better know, evaluate and apply new research methods, while the academic partners could get from industry valuable feedback, better understanding the industrial problems and needs.

Several synergies that have been established during the secondments, are now in place beyond the project termination for exploiting further potential strategic research activities. Moreover, the collaborations for the maintenance and improvement of the project tools developed during CECRIS will last for years, since these tools support the overall V&V process and reduce the certification costs of safety-critical systems.

It is the objective of this book to collect the main project results in terms of methodologies and processes and to propose them in a single edited book.

The first part of the book is related to certification processes. Chapter one presents an easy-to-use framework and a supporting methodology to perform a rapid gap analysis on the usage of standards for safety-critical software, being them new ones to be introduced or standards already applied. In other words, the framework can be applied to reason in terms of "changing standard" or in terms of "introducing a new standard". The ultimate objective is to discover with limited effort how far a company is from acquiring sufficient the necessary and sufficient level of knowledge to apply a specific standard. Our approach is based on the concept of rating the knowledge available: it starts from an understanding of the expertise of a company, and it rates the improvements, in terms of training, needed to reach an adequate level of confidence with the techniques and processes required in the standard. Our approach can be applied to an entire standard, a part of it, or to individual techniques and tools. Thus, our framework offers the possibility to depict the status of the knowledge available in the company, which may offer valuable insights on the areas that are mostly covered, and where potential improvements are possible. The approach can indicate the introduction time, which estimates the overall training time required to introduce a new standard.

The second part of the book focuses on model-driven methodologies. For a company being competitive on the market, following technologies and being updated with new trends and practices is essential. In safety-critical domains, the introduction of new practices and methodologies is slower than in other engineering fields, since safety standards and long established practices tend to defer the adoption of new emerging technologies, until

assessments and time reveal them mature and safe enough. Slow introduction of new methods is especially characterizing the railway domain where the lifespan of products could easily reach decades or even a century. Now it is long time that Model-Driven Engineering techniques and tools have been proposed, but their maturity – especially for safety-critical systems – is still debated. Some recent surveys investigated the adoption of MDE methodologies and technologies in practice. They revealed the increasing adoption of MDE in industry. The technology is attractive for the development of critical systems, since it can speed up the activities of Verification and Validation (V&V), and it enables the early verification of systems, through techniques such as model reviews, guideline checkers, rapid control prototyping and model- and software-in-the-loop Tests. These techniques shift the cost of development from the phases of V&V to the ones of requirement analysis and design, thus leading to benefits in terms of residual errors. Companies not performing model-in-the-loop testing find almost 30% more errors during module test. Chapter two reports the results of a twelve months industrial-academic partnership for the transfer of knowledge of MDE techniques from the academy to one of the company involved in the project, with the goal of assessing their level of maturity for industrial adoption. During this activity, it emerged the lack of well-defined processes for the development of a CENELEC SIL-4 safety critical signaling system that was suited for the real industrial needs.

In Chapter three focuses on the issues related to the lack of expertise in CS/OO/SysML formalisms that often lead to the need of a lot of training and support to use the modeling tools. Ideally, designers should spend all their effort on modeling and nothing else. However, existing modeling tools have lot of issues related to installation and plug-ins. The use of Google Blockly was envisaged for modeling and simulation of systems. Blockly is a visual programming library, used to model/program using interlocked blocks. Each of the blocks also supports traditional input widgets such as labels, images, textbox, checkbox, combo box, etc. It can be configured in such a way that only compatible blocks can be connected together (i.e. can be made “valid by design”). Blockly supports code and XML generation, and requires only a modern web browser which can be run on any device or operating system. However, Blockly was not readily usable for modeling using SysML/UML like formalisms. A lot of changes and customizations were made in Blockly to make it more suitable for such type of modeling.

The Third part of this book composed of Chapters four, five, six and seven, deals with V&V and quality processes.

Chapter four presents a process for finding and tackling the main root causes that affect critical systems quality. Following standards and applying good engineering practices during software development is not enough to guarantee defects free software, thus additional processes, such as Independent Software Verification and Validation (ISVV), are required in critical projects. The objective of ISVV is to provide complementary and independent assessments of the software artifacts in order to find residual defects and allow their correction in a timely manner. Independence is the most important concept of ISVV and it has been referred to and used in safety-critical domains such as civil aviation (DO-178B), railway signaling systems (CENELEC), and space missions (European Cooperation for Space Standardization – ECSS). However, such systems are still far from being perfect and it is common to hear about software bugs in aeronautics, train accidents caused by software problems, satellite systems that need to be patched after launch, and so on. This chapter presents an analysis on trends, common (and uncommon) problems and their causes, and looks at the general picture of critical defects within the software development lifecycle of space systems, considering a dataset of 1070 defects. The results are intended to help engineers in tackling the problems starting from the most frequent ones, instead of dealing with them one by one, as is traditionally done in industry nowadays. In practice, this work brings light to the main root causes of issues in space projects, which were identified, based on the defects classification and on relevant expert knowledge about those defects and about the software development process, contributing towards proposing improvements to the processes, methodologies, tools, standards and industry culture.

Chapter 5 describes a framework for automation of hazard log management on large critical projects. A hazard is any situation that could cause harm to the system or lives. Hazards depend on the system and its environment, and the probability of the hazard to cause harm is known as risk. Hazards are analyzed by identifying their causes and the possible negative consequences that might ensue. This chapter describes a modular and extensible way to specify rules for checks locally at the stake-holder side, as well as while combining data from various parties to form the hazard log (HL). The HZ-LOG automatization tool simplifies the process of hazard data collection on large projects to form the hazard log while ensuring data consistency and correctness. The data provided by all parties are collected using a template containing scripts to check for mistakes/errors based on internal standards of the company in charge of the hazard management. The collected data is then

subjected to merging in DOORS, which also contain scripts to check and import data to form the hazard log.

Chapter 6 instead deals with cost estimation for independent systems verification and validation. Validation, verification and especially certification are skill and effort demanding activities which are typically performed in an independent way by specialized small and medium enterprises. Prediction of the work needed to accomplish them is crucial for the management of such projects, which is by its very nature heavily depending on the implementation of the V&V process and its support. Process management widely uses cost estimators in planning of software development projects for resource allocation. Cost estimators use the scoring of a set of cost influencing factors, as input. They use extrapolation functions calibrated previously on measures extracted from a set of representative historical project records. These predictors do not provide reliable measures for the separate phases of V&V and certification in safety critical projects. The current chapter summarizes the main use cases and results of an activity focusing on these particular phases.

Chapter 7 addresses lightweight formal analysis of requirements which are the core items of the design (and Validation) workflow of safety critical systems. Accordingly, their completeness, compliance with the standards and understandability is a dominant factor in the subsequent steps. Requirements review is a special kind of Independent Software/Systems Verification and Validation (ISVV). The chapter presents methodologies to use lightweight formal methods supporting experts in a peer review based ISVV.

Part four of this book, composed of chapters eight and nine, deals with particular phases of V&V processes known as FMEA & FMECA.

Chapter 8 describes STECA which stands for "Security Threats, Effects and Criticality Analysis" and its application to a Smart Grids scenario. The STECA approach is meant to perform security assessment and the chapter explains the process proposed to identify vulnerabilities, their related threats, a risk assessment approach and finally a path to identify appropriate countermeasures. This process is based on the same principles used for the FMEA/FMECA process, widely used for safety critical analysis and highly regarded by the majority of international standards. STECA starts from a vulnerability point of view and moves on towards threat analysis and criticality assessment. Following the guidelines defined, the approach is then instantiated on a Smart Grid use case, resulting in a set of precise guidelines and a systematic way to perform security assessment including vulnerability evaluation and attack impact analysis.

Chapter 9 describes a composable framework support for Software-FMEA through Model Execution. Performing Failure Mode and Effects Analysis (FMEA) during software architecture design is becoming a basic requirement in an increasing number of domains. However, due to the lack of standardized early design-phase model execution, classic Software-FMEA (SW-FMEA) approaches carry significant risks and are human effort-intensive even in processes that use Model-Driven Engineering.

From a dependability-critical development process point of view, FMEA should be performed in the early phases of system design; for software, this usually translates to the architecture design phase. Additionally, for some domains, standards prescribe the safety analysis of the software architecture – as is the case e.g. with ISO 26262 in the automotive domain. Significant risk is introduced by the fact that the error propagation assumptions usually made at this stage have to hold for the final system – otherwise the constructed hazard mitigation arguments will not hold. This chapter addresses SFMEA based on a new standard for UML 2 modeling language. Throughout the chapter, the reader will be introduced to i) advances in standardized model execution semantics, ii) the outline of a composable framework built on top of executable software architecture models to help SW-FMEA, iii) a realization of such a framework applied on a case study from the railway domain.

The last part of this book, Part five, contains contributions developed in CECRIS related to Robustness and Fault injection and is composed of 3 chapters.

Chapter 10 describes a monitoring and testing framework for critical off-the-shelf applications and services. One of the biggest verification and validation challenges is the definition of approaches and tools to support systems assessment while minimizing costs and delivery time. Such tools reduce the time and cost of assessing Off-The-Shelf (OTS) software components that must undergo proper certification or approval processes to be used in critical scenarios. In the case of testing, due to the particularities of components, developers often build ad-hoc and poorly-reusable testing tools, which results in increased time and costs. This chapter introduces a framework for testing and monitoring of critical OTS applications and services. The framework includes i) a box instrumented for monitoring OS and application level variables, ii) a toolset for testing the target components and iii) tools for data storing, retrieval and analysis. The chapter presents an implementation of the framework that allows applying, in a cost-effective fashion, functional testing, robustness testing and penetration testing to web

services. Finally, the framework usability and utility is demonstrated based on two different case studies that also show its flexibility.

Chapter 11 is about the validation of a safety critical railway application using fault injection. This chapter will summarize the fault injection experiments performed with the ProSigma system. It will include a detailed description of the system, fault injection test goals, description of the fault injection tool, the results of the FI tests, etc.

Chapter 12 is concerned with robustness of complex Critical Systems. Systems are nowadays being deployed also as services or web applications, and are being used to provide enterprise-level business-critical operations. These systems are supported by complex middleware, which often links different systems, and where a failure can bring in disastrous consequences for both clients and service providers. In this chapter we present a toolset that can be used to evaluate the robustness of a given system, under the following two different perspectives: i) executing robustness tests against the service's external interface (e.g., the interface with business clients) and also inner interfaces (e.g., the application-database interface); ii) emulating the presence of source code defects, on the service middleware, to understand how the presence of a defect can affect the robustness of the overall system. The toolset has been demonstrated on a set of web services, an Enterprise Resource Planning web application, and on the popular Apache HTTP server. Results show that the toolset can be easily used to disclose critical problems in web applications and to support middleware, helping developers in building and validating more reliable services.

Although the chapters of the book are arranged in a logical order, an effort has been made to keep each chapter self-contained. This book can be used for supplemental reading for advanced teaching on Critical systems validation and verification methodologies.

Andrea Bondavalli

Francesco Brancati

List of Contributors

Alexandre Esper, *CRITICAL Software S.A., Coimbra, Portugal*

András Pataricza, *Dept. of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary*

András Zentai, *Prolan Process Control Co., Szentendrei út 1–3, H-2011 Budakalász, Hungary*

Andrea Bondavalli, *1) Department of Mathematics and Informatics, University of Florence, Florence, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica-University of Florence, Florence, Italy

Andrea Ceccarelli, *1) Department of Mathematics and Informatics, University of Florence, Florence, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica-University of Florence, Florence, Italy

Arun Babu Puthuparambil, *Robert Bosch Center for Cyber Physical Systems, Indian Institute of Science, Bangalore, India*

Cristiana Areias, *1) CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*
2) ISEC – Coimbra Institute of Engineering, Polytechnic Institute of Coimbra, Portugal

Fabio Scippacercola, *1) DIETI, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica, Italy

Francesco Brancati, *Resiltech s.r.l., Pontedera (PI), Italy*

Francesco Rossi, *Resiltech s.r.l., Pontedera (PI), Italy*

Francisco Moreira, *CRITICAL Software S.A., Coimbra, Portugal*

Gonçalo Pereira, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Henrique Madeira, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Imre Kocsis, *Dept. of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary*

Ivano Irrera, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

João Carlos Cunha, *1) CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*
2) ISEC – Coimbra Institute of Engineering, Polytechnic Institute of Coimbra, Portugal

Jorge Bernardino, *1) CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*
2) ISEC – Coimbra Institute of Engineering, Polytechnic Institute of Coimbra, Portugal

László Gönczy, *Dept. of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary*

Leonardo Montecchi, *1) Department of Mathematics and Informatics, University of Florence, Florence, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica-University of Florence, Florence, Italy

Lorenzo Vinerbi, *Resiltech s.r.l., Pontedera (PI), Italy*

Marco Vieira, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Mario Rui Baptista, *CRITICAL Software S.A., Coimbra, Portugal*

Nicola Nostro, *Resiltech s.r.l., Pontedera (PI), Italy*

Nuno Antunes, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Nuno Laranjeiro, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Nuno Silva, *CRITICAL Software S.A., Coimbra, Portugal*

Raul Barbosa, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Rosaria Esposito, *Resiltech s.r.l., Pontedera (PI), Italy*

Seyma Nur Soydemir, *CISUC, Department of Informatics Engineering, University of Coimbra, Portugal*

Stefano Russo, *1) DIETI, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica, Italy

Tommaso Zoppi, *1) Department of Mathematics and Informatics, University of Florence, Florence, Italy*
2) CINI-Consorzio Interuniversitario Nazionale per l'Informatica-University of Florence, Florence, Italy

Valentina Bonfiglio, *Resiltech s.r.l., Pontedera (PI), Italy*

Vince Molnár, *Dept. of Measurement and Information Systems, Budapest University of Technology and Economics, Budapest, Hungary*

List of Figures

List of Tables

List of Abbreviations

A/D	Analog/Digital
ALARP	As low as reasonably practicable
Alf	Action Language for Foundational UML
ASILs	Automotive Software Integrity Levels
ASPICE	Automotive SPICE
BB-PIT	Black Box Platform Independent Test Model
BB-PST	Black Box Platform Specific Test Model
BDD	Block Definition Diagram
BI	Business Intelligence
CAN	Controller Area Network
CE	Cost Estimator
CENELEC	Comité européen de normalisation en électronique et en électrotechnique
CIM	Computation Independent Model
CIT	Computation Independent Test Model
CIV	Computation Independent Viewpoint
CMMI	Capability Maturity Model Integration
COCOMO	Constructive Cost Model
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CS	Critical System
CSP	Constraint Satisfaction Problem
csp(FD)	finite-domain CSP
CTC	Central Traffic Control
Dako	Andras, need your help here
DB	Database
DI	Digital Input
DMI	*Driver Machine Interface*
DOORS	Dynamic Object Oriented Requirements System
DSL	Domain-Specific Language

Eclipse RMF	Eclipse Requirement Management Framework
ECSS	European Cooperation for Space Standardization
EER	Enhanced Entity–Relationship
EN	Européen Norme
ERTMS	European Rail Traffic Management System
ESA	European Space Agency
ETCS	European Train Control System
ETH	CAN to UDP protocol converter
FDIR	Fault Detection, Isolation and Recovery
FI	Fault Injection
FIR	Fault Injection Runs
FIT	Fault Injection Tool
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Modes, Effects and Criticality Analysis
FTA	Fault Tree Analysis
FW	Firmware
GA	Generic Application
GB-PIT	Grey Box Platform Independent Test Model
GP	Generic Product
GR	Golden Runs
GSM	Global System for Mobile communications
HA	Hazard analysis
HAN	Home Area Network
HB	HeartBeat signal
HIL	Hardware-in-the-loop
HL	Hazard log
HMI	Human-Machine Interface
HSIA	HW/SW interaction analysis
HW	Hardware
HZ	Hazard
IBD	Internal Block Diagram
ICT	Information and Communication Technology
IDEF	Integration DEFinition
IEC	International Electrotechnical Commission
IP	Internet Protocol
IS	Interlocking System
ISO	International Organization for Standardization
ISVV	Independent Software Verification and Validation

ISVV	Independent Software/Systems Verification and Validation
JIF	Relay Interface
JTAG	Join Test Action Group
KLOC	Thousands of lines of code
KPI	Key Performance Indicator
LI	Logic and Input
M2M	Model-to-Model Transformation
M2T	Model-to-Text Transformation
MBE	Model-Based Engineering
MBSE	Model-Based System Engineering
MDA	Model-Driven Architecture
MDD	Model-Driven Development
MDE	Model-Driven Engineering
MDT	Model-Driven Testing
MIL	Model-in-the-loop
MoC	Models of computation
MT	Mitigation
NIST	National Institute of Standards and Technology
OBU	On-board Unit
OCD	On-Chip Debugger
ODC	Orthogonal Defect Classification
OMG	Object Management Group
OS	Operating System
OWL	Web Ontology Language
OXF	Object Execution Framework
PA	Product Assurance
PAR	Parameter Module
PB	Prolan Block
PHA	Preliminary Hazard Analysis
PIM	Platform Independent Model
PIT	Platform Independent Test Model
PIV	Platform Independent Viewpoint
PM	Prolan Monitor
PSDK	Prosigma Diagnostic Center
PSM	Platform Specific Model
PST	Platform Specific Test Model
PSU	Power Supply Unit
PSV	Platform Specific Viewpoint
PTD	ProSigma generic application

QA	Quality Assurance
RAM	Random Access Memory
RAMS	Reliability, Availability, Maintainability, and Safety
RBC	Radio Block Control
RCA	Root Cause Analysis
RDF	Resource Description Framework
ReqIF	Requirements Interchange Format
RID	Review Identified Discrepancy
RODIN	Rigorous Open Development Environment for Complex Systems
ROI	Return on Investment
RPI	UDP to X25 over IP protocol converter
SA	Specific Application
SAM	Specific Application Module
SCAMPI	Standard CMMI Appraisal Method for Process Improvement
SDLC	Software Development Life Cycle
SDP	Software Development Process
SHA	System hazard analysis
SIL	Safety Integrity Level
SME	Small and medium-sized enterprise
SPICE	Software Process Improvement and Capability Determination
SSHA	Subsystem hazard analysis
SST	Safety Signal Transmitter
STECA	Security Threats, Effects and Criticality Analysis
SUT	System Under Test
SVF	Software Validation Facility
SW	Software
SW-FMEA	Software Failure Modes and Effects Analysis
SXF	Simple Execution Framework
SysML OMG	Systems Modeling Language
TC	Telecommand
TIU	*Train Interface Unit*
TM	Telemetry
TMR	Triple Modular Redundancy
UDP	User Datagram Protocol
UML	Unified Modeling Language

USB	Universal Serial Bus
UTP	UML Testing Profile
V&V	Verification and Validation
W3C	World Wide Web Consortium
WB-PST	White Box Platform Specific Test Model
X25	ITU-T X.25 Protocol