# SQUALE Dependability Assessment Criteria

Yves Deswarte, Mohamed Kaâniche, Pierre Corneillie, John Goodson

# SQUALE Dependability Assessment Criteria

Yves Deswarte[1]*, Mohamed Kaâniche[1], Pierre Corneillie[2] and John Goodson[3]

[1] LAAS-CNRS, 7 avenue du Colonel Roche
31077 Toulouse cedex 4, France
{Yves.Deswarte, Mohamed.Kaaniche}@laas.fr
[2] CR2A-DI, 25 quai Gallieni
92158 Suresnes cedex, France
pcorneil@cr2a-di.fr
[3] Admiral Management Services Limited, Kings Court, 91-93 High Street
Camberley, Surrey GU15 3RN, United Kingdom
goods_j@admiral.co.uk

**Abstract.** The aim of the SQUALE project is to develop assessment criteria to obtain a justified confidence that a system will achieve, during its operational life and its disposal, the dependability objectives assigned to it. The SQUALE criteria differ from traditional evaluation methods (security evaluation criteria, standards for the design, development and validation of safety critical systems), by: 1) their independence with respect to the application domains and industrial sectors, 2) their ability to address all dependability attributes, and 3) their progressiveness as a function of more or less strict requirements.

## Introduction

The increasing use of computers in all industrial sectors leads to the need to specify and design computing systems which could fulfill the requirements of the targeted applications at the lowest cost. Various requirements have to be taken into account, whether functional (accuracy of the results, response time, ease of use…) or dependability requirements such as availability, confidentiality or maintainability [1]. It is then of great importance to know if a given system, COTS or developed specifically, is able to achieve all these requirements. It is widely recognized that ensuring system compliance to functional requirements is not an easy task due to the fact that it is not always possible to check the system behavior in all possible conditions that may occur during its operational life. This is even more difficult for the dependability aspects, since it is generally not possible to exercise the system in all faulty situations for which requirements have been defined, considering not only physical faults, but also design faults, human interaction faults or malicious faults [1].

For critical applications, i.e. those for which computing system failures could cause catastrophes, it is possible to gain a sufficient confidence in the system behavior by imposing well-suited development and validation methods that are specified in sector-specific standards: railways (CENELEC EN 50126 [2], EN 50128 [3] and ENV 50129 [4]), nuclear power (IEC 60880 [5]), avionics (ARP 4754 [6] and DO 178B [7] standards), etc. These different standards share many common characteristics, which shows the need for a generic evaluation approach, such as the one considered in the IEC 61508 standard [8]. In the same way, when considering computing system security, there are evaluation criteria such as the TCSEC [9], ITSEC [10] or Common Criteria [11] that can help to assess the system ability to face possible threats. But all

* Yves Deswarte is currently on sabbatical at Microsoft Research, Cambridge, UK.

this concerns only two aspects of dependability, namely safety and security. However, it is often necessary to take into account other dependability attributes such as availability or maintainability. For instance in air or railway transportation systems, if passenger safety is essential, availability is also critical for the system profitability. It is thus of great importance to be able to check if the system achieves all its dependability requirements, not limited to safety or security.

The approach presented here has been developed within SQUALE[1] (*Security, Safety and Quality Evaluation for Dependable Systems*), a European research project which is part of the ACTS program (*Advanced Communications, Technologies and Services*). The aim of this project was to develop assessment criteria which would make it possible to gain a justified confidence that a given system will satisfy, during its operational life and its disposal, the dependability objectives assigned to it. These criteria are generic in the sense that they do not aim at a particular application sector but on the contrary they have to be general enough not to require supplementary work for the system to be evaluated and certified according to the domain standards.

## 1 SQUALE Criteria Overview

The SQUALE assessment framework and criteria incorporate some basic concepts from the security criteria and safety standards. Particularly:

- the roles of the different parties involved in the assessment process are defined: sponsor, developer, assessor;
- the notion of "target of dependability assessment" (TDA) is introduced to specify the boundaries and the scope of the assessment;
- a process oriented assessment framework defines *confidence providing activities* which aim at giving the system the functionality and the quality necessary to fulfil its dependability objectives;
- different levels of confidence are defined to grade the importance of the dependability attributes and define the objectives to be satisfied with respect to each dependability attribute;
- different levels of rigor, detail and independence are specified for the confidence providing activities as a function of the confidence levels to be achieved with respect to each dependability attribute.

## 2 The Dependability Assessment Framework

SQUALE criteria application takes into account the whole system life cycle (from the definition of concepts and initial needs to the system disposal), but it does not rely on a specific life cycle model. The SQUALE assessment framework takes into account the traditional system decomposition into subsystems and components (from the definition of the high-level requirements to the realization of system components, the assembling of these components according to the architecture, and finally the integration of the overall system into its operational environment). It is noteworthy that the system development should also include the definition of system installation, operation, maintenance and decommissioning procedures. Figure 1 summarizes the main tasks to be performed at a given level of system decomposition process

[1] Current SQUALE project partners are CR2A-DI (F), prime contractor, Admiral (UK), IABG (Germany), LAAS-CNRS (F), Matra Transport International (F) and Bouygues Telecom (F).

considered in the SQUALE assessment framework model and the *confidence providing processes* to be implemented.

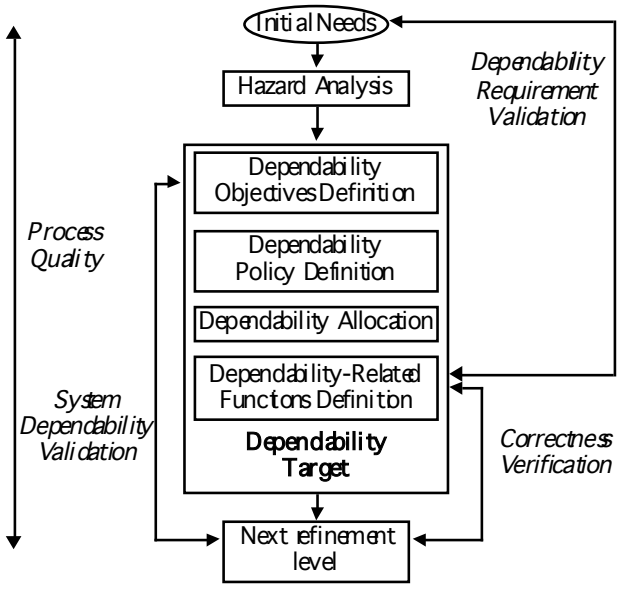


**Fig. 1.** SQUALE assessment framework and confidence providing processes

This framework is applied recursively at each refinement level of the system construction process up to the final implementation of system components and their integration. Each step of the refinement process has to start with a hazard analysis activity that consists in identifying the set of undesirable events (i.e. failures, threats, faults, etc.) that potentially have unacceptable consequences. The outputs of the hazard analysis activity should lead to: 1) the definition of the dependability objectives to be satisfied, 2) the specification of the dependability policy to be implemented to satisfy these objectives, 3) the allocation of dependability objectives to each subsystem (human, hardware, software, etc.), and finally 4) the definition of the dependability related functions to be accomplished by each subsystem.

Four *confidence providing processes* are distinguished in this general framework: dependability requirement validation, correctness verification, system dependability validation and process quality.

- *Dependability Requirement Validation* aims to ensure that at each level of system decomposition the threats and hazards of a TDA: 1) have been properly identified, 2) are covered by the respective dependability objectives, the dependability policy and the dependability related functions, and 3) comply with the initial needs of the systems.
- *Correctness verification* aims to ascertain that each level of the system implementation meets its validated requirements. One of its objectives is to check that the planned measures to prevent, tolerate, remove and forecast faults have been taken all along the development cycle and implemented correctly.
- *System Dependability Validation* checks the suitability of the dependability-related function implementation, including the effectiveness of the mechanisms implemented to counter the hazards, the ease of use of the system, the validation of

fault assumptions and the analysis of side effects of the implemented system which may lead to critical situations.

- *Process Quality* aims to ensure the correct application of the methods, tools and procedures which have to be used during all the development process, the operation and the maintenance of the system, as well as those prepared for the decommissioning phase, in order to achieve a sufficient level of quality.

## 3 Dependability target

To be assessed according to the SQUALE criteria, the system (or more precisely that part of the system which has to be assessed, i.e. the *Target of Dependability Assessment*, TDA) has to be described in a document called *Dependability Target* (similar to the Security Target document in the ITSEC). This document is intended to serve as a reference for the dependability assessment. In its initial version, the Dependability Target is developed in the earliest stage of the development process, before starting the main assessment activities. In the case of complex systems, this document has to be refined at each level of decomposition.

The Dependability Target contents are:

- the description of the system and its environment, including the system interface description (interfaces with other systems, interfaces with the physical environment, man-machine interfaces, interfaces to the organization, etc.) and the hazard-related assumptions concerning the environment, i.e. identifying the hazards which are eliminated by the environment conditions;
- the results of the hazard analysis, identifying what the system must protect and against what the system has to be protected; this analysis produces a list of threats and hazards for the system and its environment that shall be taken into account; it includes the assumptions made for each hazard in the analysis and the overall rating of the severity of each hazard with a description of the rating method;
- the definition of a set of objectives for each dependability attribute (safety, availability, confidentiality…); this activity consists in comparing the level of risks (associated to the identified hazards and threats) with what is acceptable for the system; objectives are then defined to reduce the risks to an acceptable level;
- the definition of the dependability policy which describes how to fulfill dependability objectives through high level measures which are implementation independent; these measures are composed of rules and statements that the system has to enforce; this definition produces a set of regulations, standards, practices and procedures that should be used to achieve the dependability objectives, e.g. design diversity, partitioning, fail-stop processors, etc.
- the identification of the dependability-related functions, and their specifications;
- the dependability allocation, which defines the dependability objectives and policy for each dependability-related function; its purpose is to specify the role of each subsystem (human, hardware, software, other technologies) in the enforcement of the dependability policy and objectives;
- the definition of the required *dependability profile* (see Section 4) for each dependability-related function and component;
- the dependability plan, describing confidence providing activities and methods appropriate for the TDA; the methods are chosen according to the component dependability attributes, the expected confidence level and the life cycle phase.

## 4 Dependability profile

For a given system, the non-functional requirements may apply to some or all of the dependability attributes (availability, confidentiality, reliability, integrity, safety and maintainability [1]). Moreover the importance of the attributes in a particular application may not be uniform. For instance, a safety-critical system may have safety and maintainability requirements, but the safety requirements are more significant than the maintainability requirements. In the SQUALE criteria, each attribute is assigned an expected confidence level, varying from 1 to 4, 1 being the lowest confidence level and 4 the highest one (see Table 1). A level 0 is also defined to indicate that nothing is required concerning this attribute. For instance, a system may be deemed to have a *dependability profile* A1, C0, R3, I3, S3, M2 which corresponds to confidence levels 1, 0, 3, 3, 3 and 2 respectively for availability, confidentiality, reliability, integrity, safety and maintainability.

| | |
|---|---|
| Availability | A1-A4 |
| Confidentiality | C1-C4 |
| Reliability | R1-R4 |
| Integrity | I1-I4 |
| Safety | S1-S4 |
| Maintainability | M1-M4 |

**Table 1.** Confidence levels

## 5 Confidence providing activities and assessment

The assessment consists in checking that the confidence providing activities have been selected and carried out properly to achieve the requested confidence levels, and possibly in completing them. The assessment activities are thus organized according to the four main Confidence Providing Processes (CPPs) (Figure 1). For each CPP, a set of Confidence Providing Activities (CPAs) are defined together with appropriate methods that can be used to reach the objectives of these activities:

- The CPAs corresponding to the *Dependability Requirement Validation* include the preliminary hazard analysis, the probabilistic quantitative evaluation and the common cause analysis.
- For *Correctness Verification*, static analysis, behavioral analysis, formal methods and proofs, testing and traceability analysis can be used.
- *System Dependability Validation* includes penetration analysis, covert channel analysis and experimental evaluation.
- *Process Quality* is implemented by the quality assurance activities.

For each activity, the criteria define different levels for rigor (RL), detail (DL) and independence (IL). Each of these levels may take three values, from 1 (the lowest level) to 3 (the highest level), according to the confidence level identified in the dependability profile.

The rigor level determines how the activity has to be done (e.g. from informally to formally, from manually to automatically, etc.), the degree of justification to be

provided to the assessor (e.g. suitability of methods, evaluation of tools, etc.), and the kind of evidence required (e.g. test coverage).

The detail level indicates the scope of the CPA such as whether it addresses: 1) parts or all of the system, 2) one or many refinement levels, 3) all the properties or only a subset.

The independence level specifies the organizational links between those carrying out the activity and the developers[2]. In most cases, IL1 indicates that they are independent persons, IL2 indicates independent departments, IL3 indicates independent organizations.

For each confidence providing activity, the SQUALE criteria provide a precise definition of each of the RL, DL and IL levels, as well as the relations between these levels and the confidence levels of each dependability attribute.

# 6 Complex Systems

## 6.1 Decomposition

Complex Systems are normally broken down into several subsystems, which themselves may be broken down further into sub-subsystems, and so on. Such a decomposition is necessary to:

- control the complexity of the system,
- isolate certain types of functionality (e.g. confidentiality) in a few components rather than spread it thinly throughout the system,
- allow different types of component (e.g. hardware or software) to be developed by different teams,
- allow the development of some components to be subcontracted out,
- allow some components to be implemented using Commercial Off The Shelf (COTS) products.

As a system is decomposed into components and those components are further decomposed, decisions will be taken about the allocation of requirements to each component and the design of each component. As an integral part of this process, the Dependability Framework will be applied recursively to each component. Thus, the Dependability Target will be updated with the Dependability Target information for each component. This will involve the determination of the Dependability Profile for each component. These Dependability Profiles can be (and should be) different from the system Dependability Profile. It is good practice to isolate critical functionality in a part of the system, so that its implementation can be separated from other parts of the system.

Figure 2 shows an example of a system (with a partial Dependability Profile of S4, C3, …) decomposed into 2 major components, A and B. The safety related functionality has been confined to A, and the confidentiality functionality has been confined to B, as demonstrated by their respective partial Dependability Profiles of S4, C0, … and S2, C3, … Component B has been further decomposed into components B1 and B2. B1 is slightly safety-related and confidentiality irrelevant (partial Dependability Profile of S1, C0, …) and B2 implements all the confidentiality functionality (partial Dependability Profile of S1, C3, …).

---

[2] As indicated in Section 1, the assessors should always be independent from the developers and from the sponsor.

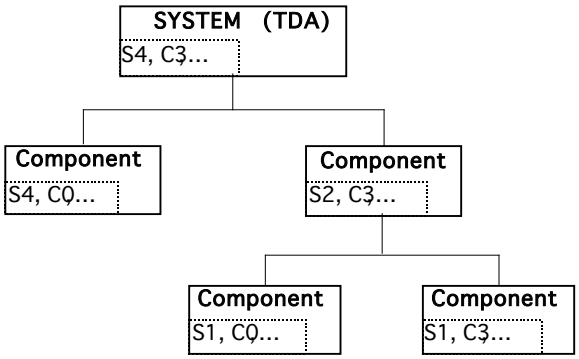


**Fig. 2.** Example of System Decomposition

In such an example, strong arguments must be provided to justify the separation of the components and demonstrate that:

- A cannot affect the confidentiality functionality of B
- B1 cannot affect the confidentiality functionality of B2
- etc.

If these arguments can be presented then consideration of the critical functionality can be confined to the relevant parts of the system. In the example shown, the activities needed for the development of a confidentiality related component can be confined to B, B2, ignoring A and B1.

## 6.2 CPAs Application.

During the system decomposition process, a hierarchy of Dependability Profiles might be defined with a Dependability Profile associated with each component. The Dependability profiles of the sub-components corresponding to a given hierarchical level, together with the associated Dependability Objectives and Dependability Related Functions should be validated taking into account the Dependability Profile, Dependability Objectives and Dependability Related Functions of the corresponding component belonging to the immediately superior hierarchical level. The validation of the dependability refinement and allocation will be based on the CPAs defined in the criteria with levels of Rigor, Detail and Independence corresponding to the dependability profile of the latter component.

If we take the example of components B, B1 and B2, the CPAs should be applied as follows:

- Use CPAs with levels of Rigor, Detail and Independence corresponding to (S2, C3) to ensure that the decomposition of B with (S2, C3) into B1 with (S1, C0) and B2 with (S1, C3) is correct and valid (i.e., satisfies the dependability requirements of component B).
- Use CPAs with levels of Rigor, Detail and Independence corresponding to (S1, C3) for B2 and (S1, C0) for B1, respectively to ensure that each component satisfies its allocated requirements.

In all cases, the CPAs must be applied with respect to each Dependability Attribute at the confidence level for that attribute. For instance, in the above example, Correctness Verification testing of component B2 needs to be performed at Confidence Level 3 for confidentiality functionality and at Confidence Level 1 for

safety functionality. Nevertheless, The developer would not be prevented from performing a CPA at a higher Confidence level than that required by the Criteria if it was thought to be more efficient. For example, in the case of component B2, the developer might decide to perform one hazard analysis at Confidence Level 3 and address both safety and confidentiality.

## 6.3 Subcontractors.

The development of a component may be subcontracted by the developer of the system (prime contractor) to another organization. That organization can be either a separate company with a formal contractual relationship or a different department or team within the same company. In the former case there will be a formal contractual relationship and in the latter case at least an implicit contract. In either case, we can regard the organization implementing the component as a subcontractor.

Irrespective of whether there is a formal contract, it is crucial that the requirements for the component are correctly and completely described. The dependability requirements will be expressed in the Dependability Target for the component.

It may or may not be possible to perform Dependability Requirements Validation and Dependability Validation against the system Initial Needs etc. depending on the prime contractor's relationship with the subcontractor. Figure 3 illustrates the two possibilities.

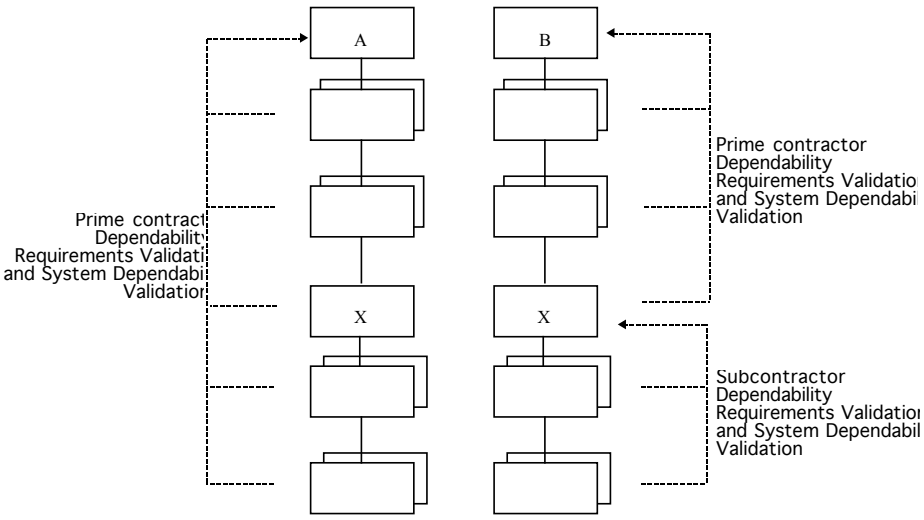


**Fig. 3.** Subcontracted Component

For system A, component X has been subcontracted but the prime contractor and subcontractor both have access to all the relevant information about the system. In this case, Dependability Requirements Validation and Dependability Validation can be performed for parts of the system (including X) against the Initial Needs etc. for the system.

For system B, again component X has been subcontracted but the relationship between the prime contractor and the subcontractor is such that:

- the prime contractor does not have access to the design etc. of X

- the subcontractor does not have access to the design etc. of the system outside component X.

This is a less desirable situation than with system A. Wherever possible, the prime contractor should try to ensure that he has sufficient access to the design etc. of X.

In this case the prime contractor can perform the Dependability Requirements Validation and Dependability Validation only on the components for which it has information. Specifically, the subcontractor can perform Dependability Requirements Validation and Dependability Validation only on the components for which it has information. Specifically, the Dependability Requirements Validation and Dependability Validation for the components of X can be performed only against the Initial Needs etc. of X, not of the system. It is most important to ensure that the requirements for X (which are imposed on the subcontractor) are correct and complete with respect to the system Initial Needs etc. Any errors or omissions will result in a component X that is not suitable for use in the system.

In each case, system A or system B, a Dependability Target is required for component X.

## 6.4 COTS Products.

There is often a need (or wish) to include COTS products in a system, for instance as component B2 in Figure 2. In order for this to happen a number of conditions must be fulfilled:

- the COTS product must provide the dependability (and other) functionality that is required of component B2;
- the COTS product must not provide too much unwanted functionality that could provide additional hazards;
- there must be sufficient confidence that the COTS product does provide just the necessary dependability functionality in the environment in which it will be used within the system;
- it must be possible to check that the previous conditions are fulfilled.

There must be a description of the COTS product identifying the functionality provided by the product and the environment(s) within which it will provide that functionality. Ideally this description will be in a Dependability Target, but it must exist, otherwise it will not be possible to check whether the product's functionality matches the requirements of the component it is to replace.

If the system component (B2 in our example) has some dependability requirements then it will have a Dependability Profile. This shows that there are some confidence requirements that have to be satisfied by the product and these confidence requirements can be obtained only by some form of assessment. There are a number of possibilities:

- the product has been subjected to an assessment against the SQUALE Criteria - there will be a certificate and assessment reports giving the results of the assessment and these results (achieved Dependability Profile, assessed functionality and environment) can be checked against the requirements for the component B2;
- the product has been assessed against some other criteria - this is similar to the previous case but it is also necessary the decide to what extent the other assessment method is equivalent to a SQUALE assessment;
- the product has not been assessed - the product cannot be used until it has been assessed either as a separate product or as part of the system assessment.

## Conclusion

The SQUALE assessment criteria have been designed to address all attributes of dependability rather than only security (like the TCSEC or the ITSEC) or safety (like the DO178B or IEC 1508). Thus, they should be useful to guide the design, development and assessment of a large range of future systems whose requirements will spread over several dependability attributes. For instance, future avionics systems should take into account malicious threats (and thus security concerns) rather than only accidental threats such as design faults, hardware failures and human errors. The same concerns apply to large infrastructure survivability. In other domains such as banks or transactional systems, security and availability are the main dependability requirements.

With respect to existing and future safety, security and quality standards, it should be easy to adapt the SQUALE criteria so that an assessment according to these criteria should be sufficient to satisfy the standard requirements, or only a little complementary effort should be needed to meet these requirements.

Moreover, the SQUALE criteria aim to be progressive: for moderate dependability requirements, the cost of the confidence providing activities and of assessment activities should be a small percentage of the development costs, and the cost of these activities should grow progressively with the dependability level to be achieved.

In order to validate and refine the first draft of the SQUALE criteria, an experiment has been carried out to evaluate the control subsystem of a new automatic subway transportation system, METEOR. Based on the results of this experiment, improvement of the SQUALE draft criteria has been undertaken towards more flexibility and efficiency. A new draft of the criteria has been published [12] and another experiment has started on a very different system: rather than an already implemented system, it is a system currently being defined by Bouygues Telecom. Moreover, the dependability requirements are much less strict for this system than for METEOR, and should concern more the security aspects than the safety aspects. We are confident that this second experiment will confirm the progressiveness and flexibility of the SQUALE criteria.

## References


1. J.-C. Laprie (ed.), Dependability: Basic Concepts and Terminology, Springer-Verlag, Dependable Computing and Fault-Tolerant Systems Series, vol.5, ISBN 3-211-82296-8, 1992.
2. CENELEC EN 50126, Railway Applications: The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS), European Committee for Electrotechnical Standardization (CENELEC), 1997.
3. CENELEC EN 50128, Railway Applications: Software for Railway Control and Protection Systems, European Committee for Electrotechnical Standardization (CENELEC), 1997.
4. CENELEC ENV 50129, Railway Applications: Safety Related Electronic Systems for Signalling, European Committee for Electrotechnical Standardization (CENELEC), 1998.
5. IEC 60880, Software for Computers in the Safety Systems of Nuclear Power Stations, International Electrotechnical Commission (IEC), 1986.
6. Certification Considerations for Highly-Integrated or Complex Aircraft Systems, ARP 4754, Society of Automotive Engineers (SAE), Nov. 1996.
7. Software Considerations in Airborne Systems and Equipment Certification, DO178B/ED12B, Radio Technical Commission for Aeronautics (RTCA), European Organization for Civil Aviation Electronics (EUROCAE), December 1992.
8. IEC 61508, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems, Parts 1 to 7, International Electrotechnical Commission (IEC), 1998-1999.

9. TCSEC: Department of Defense Trusted Computer System Evaluation Criteria, U.S. Department of Defense, 1985.
10. ITSEC: Information Technology Security Evaluation Criteria, Office for Official Publications of the European Communities, Luxembourg, 1991, ISBN 92-826-3004-8.
11. Common Criteria for Information Technology Security Evaluation, Common Criteria Implementation Board, Version 2.0, CCIB-98-026, CCIB-98-027, CCIB-98-027A, CCIB-98-028, 1998.
12. P. Corneillie, S. Moreau, C. Valentin, J. Goodson, A. Hawes, T. Manning, H. Kurth, G. Liebisch, A. Steinacker, Y. Deswarte, M. Kaâniche, P. Benoit, SQUALE Dependability Assessment Criteria, LAAS Research Report n°98456 (revised), ACTS Project AC097, Jan. 1999, 190 pages, (also available at <http://www.research.ec.org/squale/>).

**Acknowledgements.** This work has been partially supported by the European Commission as part of the ACTS Programme under project AC097. The authors are grateful to the many other participants in the SQUALE project who contributed to the work presented here, and in particular Sylvain Moreau and Claudine Valentin from CR2A-DI, Alan Hawes and Tim Manning from Admiral, Helmuth Kurth, Götz Liebisch and Angelika Steinacker from IABG, and Paul Benoit from Matra Transport International.