DOT/FAA/TC-16/18

Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport
New Jersey 08405

Final Report for Software Service History and Airborne Electronic Hardware Service Experience in Airborne Systems

November 2016

Final Report

This document is available to the U.S. public through the National Technical Information Services (NTIS), Springfield, Virginia 22161.

This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof. The U.S. Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the objective of this report. The findings and conclusions in this report are those of the author(s) and do not necessarily represent the views of the funding agency. This document does not constitute FAA policy. Consult the FAA sponsoring organization listed on the Technical Documentation page as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

1. Report No.
DOT/FAA/TC-16/18

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
FINAL REPORT FOR SOFTWARE SERVICE HISTORY AND AIRBORNE ELECTRONIC HARDWARE SERVICE EXPERIENCE IN AIRBORNE SYSTEMS

5. Report Date
November 2016

6. Performing Organization Code

7. Author(s)
Laurence H. Mutuel, Franck Bayle, Franck Aime

8. Performing Organization Report No.
D8

9. Performing Organization Name and Address
Thales Avionics, Inc.
2733 South Crystal Drive, Suite 1200
Arlington, VA 22202

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
DTFACT-13-D-00008

12. Sponsoring Agency Name and Address
Federal Aviation Administration
William J. Hughes Technical Center
Aviation Research Division
Atlantic City International Airport, NJ 08405

13. Type of Report and Period Covered
Final Report

14. Sponsoring Agency Code
AIR-130

15. Supplementary Notes
The Federal Aviation Administration William J. Hughes Technical Center Aviation Research Division Technical Monitor was Srini Mandalapu.

16. Abstract
RTCA DO-178C "Software Considerations in Airborne Systems and Equipment Certification" and RTCA DO-254 "Design Assurance Guidance for Airborne Electronic Hardware" acknowledge, respectively, that acceptable justification software service history and airborne electronic hardware (AEH) service experience may be used as an acceptable alternative to satisfy one or several safety objectives. The selected approach for using such data is to provide an equivalent level of confidence in the software or AEH maturity (or stability) to claim certification credit for a product with usage history but which is not initially compliant with these standards. It also applies to modified products for which standard process activities would not be the best option and as part of the selection process for a new development. This research is applicable to equipment, hardware, and software items.

This research proposes two approaches: bottom-up and top-down. The bottom-up approach furthers the current guidance of software and hardware levels using qualitative and quantitative criteria for determining the suitability of the collected data in service. Multistep question-based decision diagrams are proposed to support the determination by the applicant/the designee of the applicability of the claimed certification credit. Qualitative criteria such as criticality, level of innovation, complexity, and impact of the fault on the user are assessed through more detailed sets of questions for which any negative answer should lead to reconsidering the suitability of the data for the intended objective. The similarity in operating environment is evaluated from investigating potential qualitative and quantitative differences in usage domain and environmental conditions. The quantitative criteria focus on obtaining and justifying reliability estimates using reliability modeling, for which the selection of a model that meets prescribed performance objectives is key. Though the reliability of software and AEH are different, the selection of a reliability model based on statistical properties is applicable to both domains.

The primary limitation of the bottom-up approach is that it requires a significant amount of detailed information that may not be available through the data collection process and the proposed decision flows are restrictive on the usability of the data. The top-down approach investigates service history at a system level and abstracts safety goals and system descriptions into an Equivalent Design Assurance Level (EDAL) that can be directly linked with a minimum service history period. This EDAL, together with an estimated level of safety, allows for claiming an equivalent level of safety. The recommended minimum service history is based on Title 14 Code of Federal Regulations Part 25 statistics, but should be transferable to other parts and is applicable to all design assurance levels.

17. Key Words
Software service history, Product experience, Reliability, Environment, Data collection, Certification credit, DO-178C, DO-254, DO-278A, Design assurance, Equivalent level of safety, Aircraft software, Airborne electronic hardware.

18. Distribution Statement
This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161. This document is also available from the Federal Aviation Administration William J. Hughes Technical Center at actlibrary.tc.faa.gov.

19. Security Classification (of this report)
Unclassified

20. Security Classification (of this page)
Unclassified

21. No. of Pages
195

22. Price

ACKNOWLEDGEMENTS

To Cyril Marchand for the approach to software service history and Guy Berthon and B. Durand for their review of the findings and proposed recommendations.

To the Aerospace Vehicle Systems Institute Authorization for Expenditure 83 project team for allowing access to its reliability model. In particular, we wish to thank Charles Kilgore, of the Federal Aviation Administration, and Lori Bechtold, of The Boeing Company, for the engaging discussions that were ultimately captured in the recommendations for further research section of this report.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AC	Advisory Circular
AEH	Airborne electronic hardware
AFE	Authorization for Expenditure
AMC	Acceptable Means of Compliance
AMSAA	US Army Materiel Systems Analysis Activity
ANS	Air Navigation Services
ARP	Aerospace Recommended Practice
ATM	Air Traffic Management
AVSI	Aerospace Vehicle Systems Institute
BAO	Bad as old
CAST	Certification Authorities Software Team
CAT	Category (problem report classification)
CFR	Code of Federal Regulations
CM	Certification Memorandum
CNS	Communication, navigation, surveillance
COTS	Commercial off-the-shelf
CSM	Common safety method
CTMC	Continuous-time Markov chain (model)
D	Documentation
DAL	Development Assurance Level (hardware, RTCA/DO-254 context)
DAL	Design Assurance Level (software, RTCA/DO-18B context)
DBPLP	Doubly Bounded Power Law Process
DO	Delivery Order
EASA	European Aviation Safety Agency
EDAL	Equivalent Design Assurance Level
EFH	Equivalent flight hours
ELOS	Equivalent level of safety
ELS	Estimated level of safety
EOC	Executable object code
EUROCAE	European Organisation for Civil Aviation Equipment
FDA	Food and Drug Administration
FRACAS	Failure reporting, analysis and corrective action system
GAN	Good as new
GM	Geometric Moranda (model)
GO	Goel-Okumoto (model)
GPF	Generalized power family (model)
HE	Hyper-exponential (model)
HLR	High-level requirement
HPP	Homogeneous Poisson process
JM	Jelinski-Moranda (model)
KHO	Khoshgoftaar (model)
LLR	Low-level requirement
LP	Log power (model)
LRU	Line replaceable unit
LSE	Least square estimator

MBCA	Model based on control abstraction
MBIA	Model based on information abstraction
MBSE	Model-based system engineering
MER	Magnitude of error relative to the estimate
MLE	Maximum likelihood estimator
MMER	Mean magnitude of error relative to the estimate
MMRE	Mean magnitude of relative error
MRP	Markov renewal processes
MSE	Mean square error
MTBF	Mean time between failures
MTTF	Mean time to failure
NHPP	Non-homogeneous Poisson process
O	Ohba (model)
P	Problem reporting
PHAC	Plan for Hardware Aspects of Certification
PL	Perception Level
PLP	Power Law Process
PSAC	Plan for Software Aspects of Certification
R	Relevance
S	Sufficiency
SEE	Single event effects
TIAM	Technology Independent Assurance Method
TSO	Technical Standard Order
YOO	Yamada-Ohba-Osaki (model)

EXECUTIVE SUMMARY

RTCA Delivery Order (DO)-178C “Software Considerations in Airborne Systems and Equipment Certification” and RTCA DO-254 “Design Assurance Guidance for Airborne Electronic Hardware” acknowledge, respectively, that with acceptable justification, software service history and airborne electronic hardware (AEH) service experience may be used as an acceptable alternative to partially satisfy safety objectives relevant to the development/design level of an aircraft product. Five questions frame the research:

1. When to use software history or AEH service experience to support certification credit?
2. What criteria should be used to assess the relevance of software history or AEH service experience data?
3. What criteria should be used to assess the sufficiency of software history or AEH service experience?
4. What considerations should be required regarding problem reporting?
5. What information should be provided in the supporting documentation for certification?

Based on the current guidance standards, a bottom-up approach was developed to support the applicant and the authority in determining whether the data collected in-service can be used to claim certification credit for software and hardware items. The selected approach for using software service history or AEH service experience was aimed at increasing confidence in the maturity (in a reliability sense) of the software or AEH; this confidence supports the claim for certification credit in the following instances:

- A software or hardware product with usage history that was not certified in compliance with RTCA DO-178C or RTCA DO-254.
- A modified software or hardware product for which data to satisfy some or all of the RTCA DO-178C or RTCA DO-254 objectives may not be available.
- The selection process of a software or hardware product as part of a new development that will be approved in compliance with RTCA DO-178C or RTCA DO-254.

A second approach, top-down, placed the focus at system level to provide more directly actionable answers to the framing questions. Service history may be considered for use if an equivalent level of safety (ELOS) can be demonstrated by the applicant. This ELOS can be claimed using an equivalent level of design assurance in conjunction with an estimated level of safety. The equivalent level of design assurance can be directly tied to a minimum period of service history and system-level information collected within a prescribed perception level.

The approaches in this report can be used with any kind of product in mind, be it a full unit of equipment such as a line replaceable unit; a hardware circuit board assembly; an individual hardware device or component; or a piece of software such as a module or library. The product may also be developed for any original environment or domain (avionics or non-avionics), or it may be commercial off-the-shelf or a previously developed in-house hardware or software item. Both approaches highlight issues with the current data collection process and the problem reporting to ensure that the appropriate level of detail is extractible from the data, and that failure conditions are appropriately classified. This impacts the design of the maintenance system, problem-reporting process, and change analysis process.

The bottom-up approach resulted in proposed multistep question-based decision diagrams to support the determination by the applicant/authority of the suitability of data collected in-service with respect to the claimed certification credit. The suitability is determined via the analysis of a series of both qualitative and quantitative criteria. Qualitative criteria such as criticality, level of innovation, complexity, and impact of the fault on the user are assessed through more detailed sets of questions for which any negative answer should lead to reconsidering the suitability of the data for the intended objective.

The similarity in operating environment is evaluated by investigating potential qualitative and quantitative differences in the usage domain and environmental conditions. The multistep approach allows for determining whether differences related to usage domain should lead to the disqualification of the product service experience. The approach covers product configuration (e.g., part number and impact of change in part number), external interfaces (e.g., input/output configuration and impact of change in input/output), functional configuration (e.g., installation changes and operating modes), and the exchange of data between the product and external systems (e.g., changes in allowable input range and changes in data exchange rate). The consideration of environmental conditions relates to RTCA DO-160 document "Environmental Conditions and Test Procedures for Airborne Equipment" and inquires whether the target environment would be more severe than the in-service environment; in such a case, additional endurance test data may be required to supplement the product service experience.

The quantitative analysis focused on the provision of reliability estimates (e.g., mean time between failures) using reliability models to be compared with the reliability objective sets by the original equipment manufacturer. Though the reliability of software and AEH are different, the selection of a reliability model based on statistical properties is applicable to both domains. Numerous models are available in the literature, and a selection of models being used for aerospace applications is included in this report—therefore, the difficulty is primarily on the selection of such a model for the intended objective. The research presents several quality-of-fit metrics that can be used to demonstrate the suitability of the model; these metrics relate to the minimum amount of service history through performance objectives, to be agreed upon by the community.

The bottom-up approach requires access to more information than the top-down approach but is the one currently being used when software service history or AEH service experience is being used. Its limitations and difficulty level are major reasons for the limited use of this alternate means of compliance. Most decision paths with this approach lead to discarding service history. Conversely, the top-down approach is straightforward but requires further research to recommend practical implementation of the equivalent level of design assurance.

This report concludes with three open discussion items: the need for an agreed upon scheme to classify faults; investigation of suitable methods other than reliability modeling for the assessment of product stability and the improvement of reliability in the design phase; and comparative assessment of the safety net approach proposed for the selection of microprocessors and the decision criteria recommended in this research.

1. INTRODUCTION

1.1 PURPOSE

This report is produced under an FAA research contract as part of the FAA's exploration of various alternative approaches to software development assurance and airborne electronic hardware (AEH) design assurance to enhance and streamline the aircraft certification process.

The primary focus of this report is to determine the criteria to be applied on software service history and AEH service experience to support a claim of certification credit based on that data. The report provides insight into the current difficulties of collecting such suitable data and addressing the guidelines documented in the most current standards for aircraft software and hardware assurance. Several decision-control flow diagrams are provided for the practitioner so they may evaluate whether or not the data collected are acceptable.

1.2 SCOPE

The scope of this report and the underlying research focuses on software service history used in airborne applications and AEH service experience as acceptable alternative methods to claim certification credit. As a means to draw comparison, standards from other domains are discussed, including Air Traffic Management (ATM) applications, software used in rail transportation critical systems, software used in nuclear safety critical applications, and software used in safety critical medical applications.

This report covers both aircraft software and AEH, and focuses on common issues; when a differentiation is required for clarity of language, the text specifies the applicability to software only and identifies the differences with AEH considerations on that particular matter.

1.3 BACKGROUND

Software service history is indicated in RTCA Delivery Order (DO)-178C, "Software Considerations in Airborne Systems and Equipment Certification" [1], as an acceptable alternative method to comply with software assurance processes, and AEH service experience is included in RTCA DO-254, "Design Assurance Guidance for Airborne Electronic Hardware" [2], as an alternative method to comply with hardware assurance processes. In both cases, the criteria under which software history data or hardware service experience are acceptable to claim certification credit are not quantified. It is therefore agreed upon on a case-by-case basis between the applicant and certification authority. The confidence in the in-service data and their use in models to estimate reliability are key elements to support the decision-making process.

Though software does not fail in the same sense that electronic hardware does, the criteria for accepting lifetime data share similarities. Therefore, the models used to estimate their respective reliability are similar. This research considered software applications and electronic hardware for the aerospace industry and other domains as comparison points.

1.4 RELATED ACTIVITIES AND DOCUMENTS

The following documents relate directly to the issues addressed within this report:

- DOT/FAA/AR-01/116, “Software Service History Handbook” [3] and DOT/FAA/AR-01/125, “Software Service History Report” [4] contain previous research in the context of RTCA DO-178B, “Software Considerations in Airborne Systems and Equipment Certification,” which this report updates and extends.
- RTCA document DO-178C is the reference standard document used to discuss aircraft software safety assurance processes. This document recognizes that with acceptable documentation, service history may be used to support a claim as an acceptable alternative method. This document is recognized via FAA Advisory Circular (AC) 20-115C [5] and the European Aviation Safety Agency’s (EASA) Acceptable Means of Compliance (AMC) 20-115C [6].
- RTCA document DO-254 is the reference standard document used to discuss AEH safety assurance processes using service experience as an acceptable alternative method.
- RTCA document DO-278A, “Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems” [7], is the standard document used to compare processes and criteria for communication, navigation, surveillance (CNS)/ATM applications. The European Organisation for Civil Aviation Equipment (EUROCAE) document, ED-153, “Guidelines for ANS Software Safety Assurance” [8], is an older document still in use across Europe for Air Navigation Services (ANS) software applications.
- RTCA document DO-248C, “Supporting Information for RTCA DO-178C and RTCA DO-278A” [9], for the discussion papers and clarifications added to the material on service history.

For the purpose of comparing assurance standards for the aviation domain, the following documents were analyzed:

- FAA’s AC 20-148 “Software Reusable Components” [10].
- Certification Authorities Software Team (CAST) position paper #27, “Clarification on the Use of RTCA Document RTCA DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware” [11], CAST position paper #29 “Use of COTS Graphical Processor (CGP) in Airborne Display Systems” [12], and CAST position paper #31 revision 4 “Technical Clarifications Identified for RTCA DO-254 EUROCAE ED-80” [13].
- FAA Order 8110.105 change 1, “Simple and Complex Electronic Hardware Approval Guidance” [14] and FAA Order 8110.49 change 1, “Software Approval Guidelines” [15].
- EASA Certification Memorandum (CM) CM-SWCEH-001, issue 1, revision 1, “Development Assurance of Airborne Electronic Hardware” [16] for its complementary information and guidance to be used in conjunction with DO-254, and EASA CM-SWCEH-002, issue 1, revision 1, “Software Aspects of Certification” [17], for its complementary information and guidance to be used in conjunction with DO-178C.

The reference documents of other safety domains analyzed for comparison with the aviation domain, as an extension to the work documented in the FAA report on service history, included:

- For nuclear industry: IEC standard 60880 “Nuclear Power Plants–I&C Systems Important to Safety–Software Aspects for Computer Based Performing Category A Functions” [18].
- For safety related domains in general: IEC standard 61508 “Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. Part 3–Software Requirements” [19].
- For medical devices, the update to the Food and Drug Administration (FDA) guidance “Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software Use in Medical Devices” [20].
- For rail applications in the United States, Title 49 Code of Federal Regulations (CFR) (transportation) chapters II (Federal Railroad Administration) and VII (National Railroad Passenger Corporation). Hardware and software considerations can be found under part 229 subpart E (locomotive electronics) and part 236 subpart H (control systems) [21].
- For rail applications in Europe, the policy for railway safety and the process for mutual recognition between member states as specified in the European Directive 2008/110 “Railway Safety Initiative,” augmented by the entry into force of the common safety methods (CSMs) applicable at the national level. The state-level example is the United Kingdom (U.K.), whose guidance on CSM is documented in the office of rail regulation’s guidance on the application of the CSM on risk evaluation and assessment.

Some of the open discussion items were based on exchanges with the Aerospace Vehicle Systems Institute (AVSI) Authorization for Expenditure (AFE) 83 project on semiconductor reliability and based on the following documents:

- AFE 83 user’s guide version 1.2
- Semiconductor reliability model version 1

2. APPROACH

This section describes the approach used to obtain the research findings and provides the definitions and synopses of methodologies necessary to understand the development of the recommendations.

Unless specified otherwise, the approach in this report can be used with any kind of product in mind, be it a full unit of equipment such as a line replaceable unit (LRU); a hardware circuit board assembly; an individual hardware device or component; or a piece of software such as a module or library. The product may also be developed for any original environment or domain (avionics or non-avionics), or it may be commercial-off-the-shelf (COTS) or a previously developed in-house hardware or software item. In brief, the term “product” is used in its broadest sense.

2.1 QUESTION-BASED APPROACH FOR APPLICANT AND AUTHORITY

Based on the analysis of the industry standards that are used as an AMC with airworthiness requirements and certification specifications, five discriminating elements were defined and expressed to help the applicant decide whether or not to use this alternative method and for the authority to assess the application data:

1. When and how to use service history/product experience to support certification credit?
2. What are the criteria to assess the relevance of service history/product experience data?
3. What are the criteria to assess the sufficiency of service history/product experience period?
4. What considerations should be required regarding problem reporting?
5. What information should be provided in the supporting documentation for certification?

Each element of the answers extracted from the standards and related documents received an identifier associated with each of the above bullets, in order: (U) for criteria related to use, (R) for criteria related to relevance, (S) for criteria related to sufficiency, (P) for considerations related to problem reporting, and (D) for documentation-related requirements.

2.2 BOTTOM-UP APPROACH: QUALITATIVE AND QUANTITATIVE CRITERIA

Service data used to claim certification credit should be analyzed both qualitatively and quantitatively to cover the criteria of relevance and sufficiency. Qualitative data are necessary to evaluate the type of quantitative data that may be used to build the product service experience.

This approach is considered to be bottom-up because it takes into account the data collection process and makes assumptions on the data collected. Some of the investigations will further require access to the product's architecture (physical and functional). The quantitative criteria include reliability modeling, which itself makes an assumption on the product and the data collection process. Finally, the criteria require that hardware and software be distinguished.

Currently, this approach is difficult to implement. The application of the decision criteria presented in this report leads, in most cases, to an assertion that most of the service history or product experience will not be suitable to use for certification credit. A restrictive case of suitability is possible for Design Assurance Level (DAL) D software and Development Assurance Level (DAL) D hardware.

Conversely, the proposed top-down approach is applicable to all development assurance levels and does not distinguish between hardware and software items.

2.3 TOP-DOWN APPROACH: BLACK-BOX SYSTEM AND EQUIVALENT LEVEL OF SAFETY

To provide a direct answer to the five framing questions, service history may be considered for use if an equivalent level of safety (ELOS) is provided by the applicant. This top-down approach proposes qualitative and quantitative criteria based on the definition of Equivalent Design Assurance Level (EDAL) and the products' estimated level of safety (ELS).

For all levels of development assurance, the EDAL is associated with in-service data for suitability and relevance. The EDAL, together with ELS, allows for claiming an ELOS.

Though similar to the proposed Technology Independent Assurance Method (TIAM) in [22], this approach is novel, clear, and simple. It uses principles borrowed from system control theory and cybernetics. While the method is not yet implemented within certification projects—as this would require modifications to the current guidelines—it is free of the limitations and complexities of the bottom-up approach.

3. BACKGROUND ON COLLECTING DATA

To claim some level of certification credit for a product based on its service history, it is necessary to have collected data for a defined period of time and demonstrated the proper functioning of the product during this same evaluation period. To be credible and valid, this data must have been collected within an environment that is at least shown to have been similar to the intended target environment. This section discusses several aspects related to the data and the collection process, including ways to characterize the data, the systems that produced them, and the faults that may be captured during the collection. Typical types of data currently collected in service are itemized, and difficulties related to the usability of the collected in-service information are presented.

These difficulties have a direct impact on the practical applicability of the bottom-up approach criteria presented in sections 4 and 5. One of the advantages of the top-down approach presented in section 6 is that its criteria are free of these limitations.

3.1 DATA COLLECTION PROCESS

The objectives of the data collection process must align with the approved strategy regarding the use of the data within the certification project. Whether in the development phases or in-service operations, the data collection process aims at gathering valid data, avoiding redundant messages, filtering events to detect and classify faults (in particular to identify safety events), and maintaining an accurate and up-to-date status on the product reliability throughout its operational life.

To capture in-service events from the end-user (e.g., an airline), the data collection process is based on this end user’s capability to provide these events to the manufacturer and on the manufacturer’s capability to process (i.e., filter) and correctly analyze these events. The more detailed the events, the more relevant the corrective actions, statistical measures, and reliability models. As aircraft operations in the aeronautical civil domain are assumed to be similar from one airline operator to another, the in-service data set is made to be rather homogeneous at the airline collection level, which is the first link in the data collection chain. The main drawback of this homogeneity from the start is that the data collection process may therefore be unified across products and users. For a same aircraft make and model, aircraft avionics systems may vary in terms of manufacturers—as buyer-furnished equipment instead of supplier-furnished equipment. This artificial homogeneity works against the achievement of a greater level of details, which is key to reliability assessment using in-service data.

3.1.1 Types of Collected Data

For most lifetime data, exhaustive life tests are difficult to implement when the mean time to the target event or time in between target events is large (e.g., death). Therefore, truncated (or censored) life tests are common. A life test consists of selecting a random sample of X components, testing them under specific environmental conditions, and observing the time to failure of each component. Though failure events relating to reliability investigations may happen more often, reliability data sets are truncated too. The condition for truncation and collection of data impacts the analysis and model fitting processes [23].

In the context of reliability with mean time between failures (MTBF) as its measure, the types of failures to be considered are the ones consistent with the MTBF context, namely all conditions wherein the system no longer performs its intended functionalities. Though software and hardware fail differently, the application of life test is valid for both. The subtypes may include anomalous behavior, loss of function, loss of operation, diminished system performance, etc.

3.1.1.1 Sample-Truncated Data

Sample-truncated data or failure-truncated data are collected when the test is terminated after the first N failures have occurred, regardless of the time it took to reach that number. Figure 1 provides a sketch of a sample-truncated data collection. In this case, the number of failures is a given parameter of the life test, while the test duration is random.

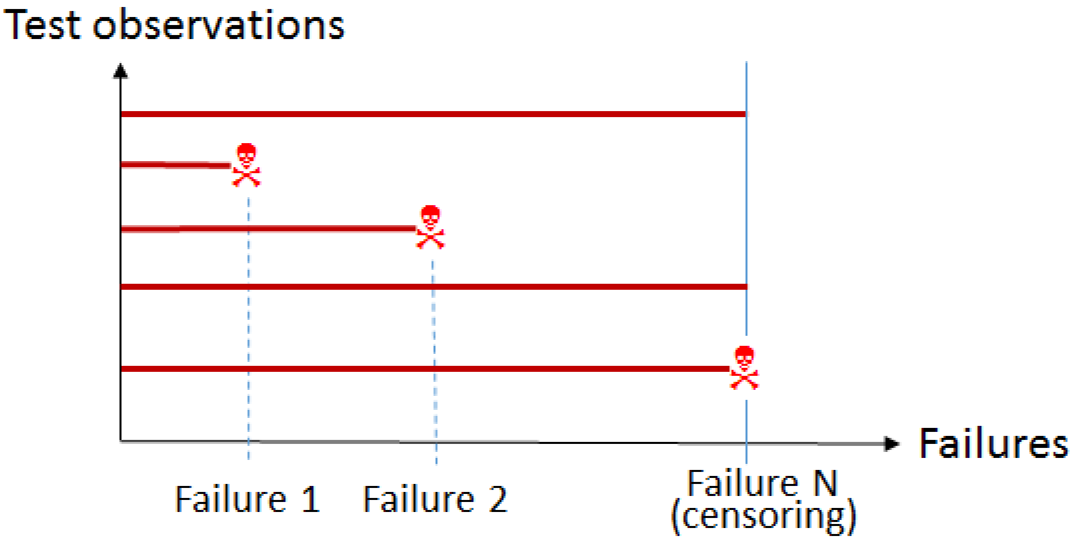

Figure 1. Sketch of failure-truncated data

3.1.1.2 Time-Truncated Data

Time-truncated data are collected when the test is terminated after a specific time (also called censoring time) has elapsed, regardless of the number of failures that occurred during the test. figure 2 provides a sketch of time-truncated data. In this case, the test duration is a given parameter of the life test, and the number of failures is random.

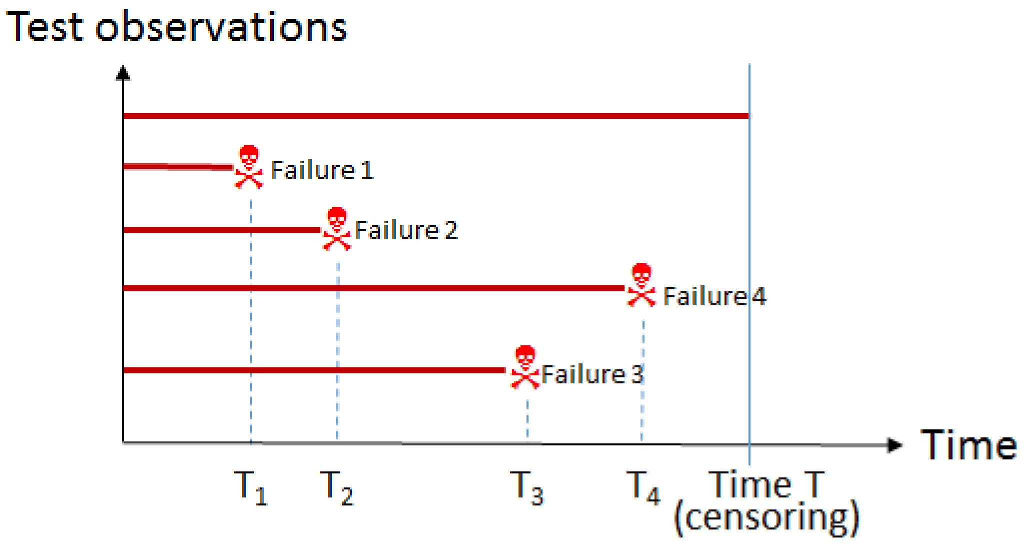

Figure 2. Sketch of time-truncated data

3.1.1.3 Grouped Data

Grouped data contain failures for which the exact time of occurrence is not known; rather, the occurrence of the failures is provided in the form of a time interval. This type of data is often used because the collection of failure information is easier (e.g., a monthly batch). For a mature product, the intrinsic vagueness in the failure occurrence time does not matter, as the underlying failure is catalectic (i.e., memoryless).

The qualifier of "memoryless" is given to probability distributions or stochastic processes for which the past has no bearing on the future behavior. Every instant is equivalent to the beginning of a new random period, regardless of how much time has elapsed. In this context, the probability of a failure to occur is the same (definition of catalectic failure), regardless of how long the system has been running.

3.1.2 Failure Reporting

A failure reporting, analysis and corrective action system (FRACAS) is a closed-loop feedback path in which the user and the supplier work together to collect, record, and analyze failures of both hardware and software data sets [24]. The airline user captures predetermined types of data pertaining to all problems associated with a particular hardware or software and submits the data to the supplier's repair center as entry into a specific database. The data are typically provided in the form of a monthly batch drop.

Following the analysis of an event report, a problem report may be raised on the product for either an immediate corrective action or for tracing this event as a known limitation of the product for a restricted period of time (i.e., until an opportunity of correction occurs). The data obtained from a FRACAS represent manufacturing quality, both in terms of fabricated products and production processes, and contain various company approvals. Corrective actions and product updates are decided on at regularly scheduled control board meetings. The main difficulties with data captured in-service are discussed in section 3.4.2.

Throughout the software life cycle (from development to verification phases and during in-service operation), problems are regularly reported into problem management tools. The identification of a problem is performed based on its description, analysis of root causes, classification, means for detecting the issue, etc. For example, numerous fields are to be filled using a tool such as ClearQuest$^{®}$. The following are examples of such fields to be entered in problem-reporting tools:

- The consequence(s) of a fault
- The root cause(s) of a fault
- The functional structure of the software
- The means of evaluating an average execution time of the software functions or operational states based on the software static and dynamic architecture

The complexity of certain software components and the potential impact on timing of abnormal conditions may complicate the determination of such average operating times. However, new information is generally obtained anyway and, despite being approximate, may be usable.

3.2 CLASSIFICATION OF SYSTEMS AND MAINTENANCE ACTIONS

This classification of systems relates to the reliability domain and allows for the use of distinct statistical descriptions and different tools. As a side effect, software is isolated to one class while hardware, depending on its expandability, may belong to both.

3.2.1 Non-Repairable and Repairable Systems

A non-repairable system is by definition not repaired and discarded after its one and only failure, as shown in figure 3. The system in this case would be discarded after failure 1.

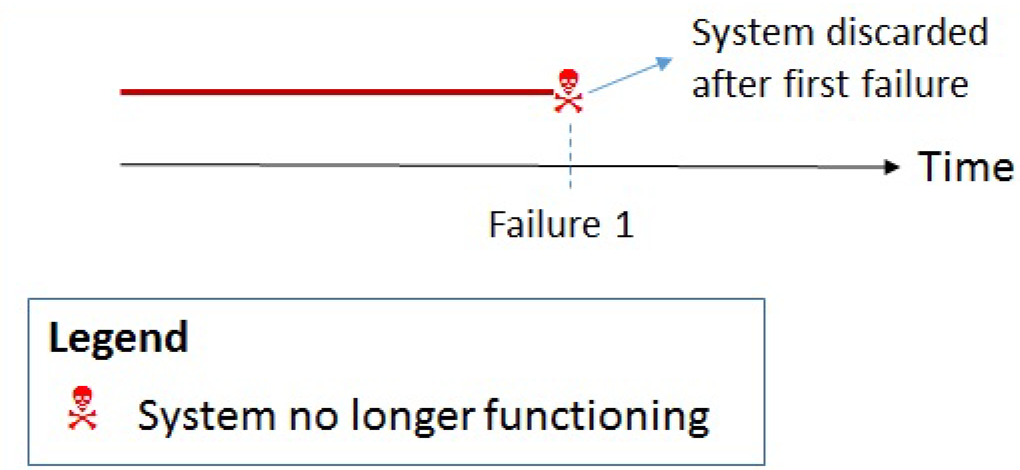

Figure 3. Non-repairable system lifetime sketch

Non-repairable systems or products are also called “one-shot” and include simple devices, such as lightbulbs, to more complex systems such as pacemakers. Reliability for these items is associated with the failure rate function, which defines the anticipated frequency with which the item will fail.

Conversely, a repairable system is a system that can be restored after a failure has occurred to an operating condition by some repair process other than replacement of the entire system, as shown in figure 4. After the occurrence of failure 1, a repair restores the system to an operating condition until failure 2 occurs; the cycle repeats with system in operation followed by the occurrence of a failure requiring repair.

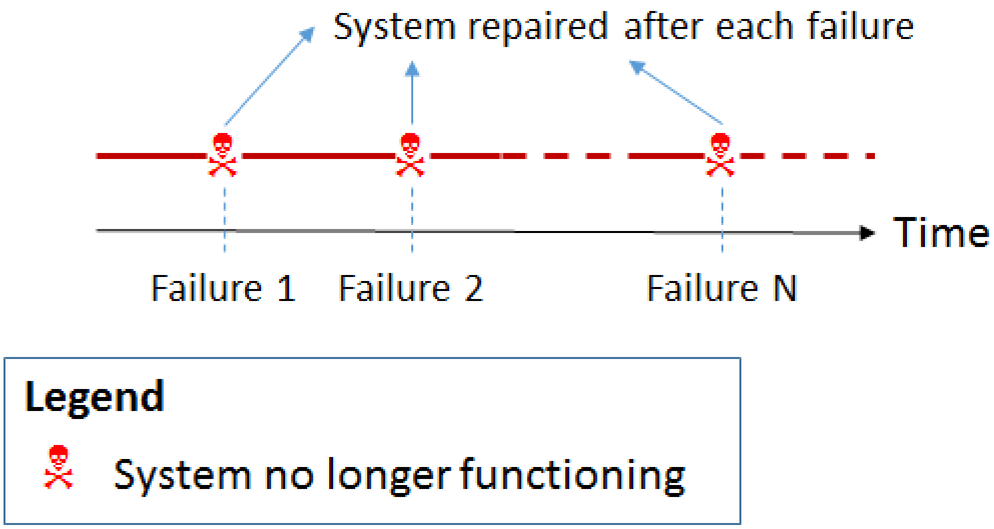
System repaired after each failure
Time
Failure 1
Failure 2
Failure N
Legend
☠ System no longer functioning

Figure 4. Repairable system lifetime sketch

Most systems can be classified as repairable systems (e.g., computers, automobiles, and aircraft), and most repairable systems include, down to some level, non-repairable parts. Of notable interest, software belongs to the class of repairable systems for which several failures can occur during its service life. The reliability of repairable items is associated with failure intensity, which is defined as the anticipated number of times the item will fail in a specified time period, given that it was as good as new (GAN) at time zero and is functioning at time *t* (see glossary, section 8).

Because of the unavailability of a component, a repairable system may become temporarily non-repairable. These transitions in classification should be accounted for when discussing the relevance of service history data and the selection of reliability models.

To illustrate a superficial comparison between repairable and non-repairable systems, table 1 provides an overview of some metrics and descriptions, which are further developed in the subsequent sections of this report—with a focus on repairable systems and products.

Table 1. Quick comparison of repairable and non-repairable systems

Metrics/ Description Item	Non-repairable System	Repairable System
Action after failure	Discard	Restore to operating condition without full replacement
Time to Failure	Mean time to failure, time to first failure, hazard rate	MTBFs, rate of occurrence of failures
Maintainability	N/A	Downtime
Lifetime description	Random variable described by single time to failure; systems are grouped, lifetime is assumed independent and identically distributed	Age of the system or total hours of operation; random variables of interest are times between failures and number of failures at a particular age
Analysis methods:		
• Weibull	Useful (single failure mode)	Not used at system level
• Reliability growth	Usually not used	Used during development testing
• Point processes	Homogeneous Poisson process	Non-homogeneous Poisson process
• Mean cumulative function	Usually not used	Useful but non-parametric

3.2.2 Types of Maintenance Actions and Impact on Data

For repairable systems, availability is important. When the required functions cannot be performed because a failure has occurred, the faster the system can be repaired the sooner its availability is restored. Therefore, maintainability is another key element to be considered. In particular, the type of maintenance, whether preventive or corrective, has an impact on the system's failure intensity in time [25]. To frame the discussion, the impact of three types of maintenance is analyzed.

Maintenance is associated with repairable systems. For comparison, the failure rate of a non-repairable system is a continuously increasing function: as time passes, the probability of the occurrence of a failure increases.

3.2.2.1 Perfect Maintenance

Perfect maintenance maintains or restores the condition of a component to GAN condition. In the context of a repairable system, consider for example that the system is exchanged for a new one after each time it has failed. Though the failures occur randomly in time, each repair "resets" the conditional probability of failure that is represented by the failure intensity. Figure 5 shows the shapes of the failure rate and the failure intensity functions.

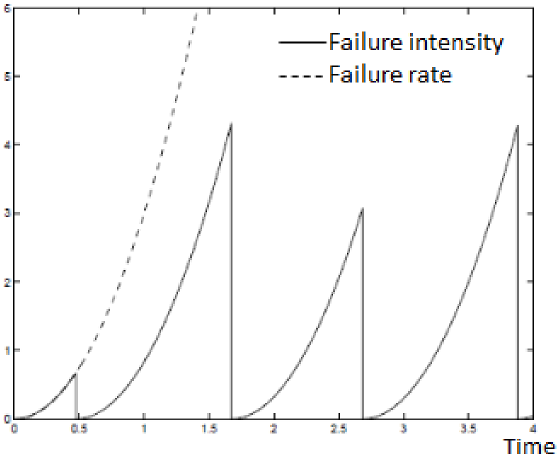

Figure 5. Failure rate and failure intensity for perfect maintenance

This type of maintenance does not account for any wear-out phenomena. It may be plausible for systems with one structurally simple component, or it represents the replacement of a failed system by a new one.

3.2.2.2 Minimal Maintenance

In this case, the maintenance performed on a system leaves the system in the exact same reliability level it was just before the failure occurred—that is, bad as old (BAO). The shape of the failure intensity (and failure rate) function is shown in figure 6.

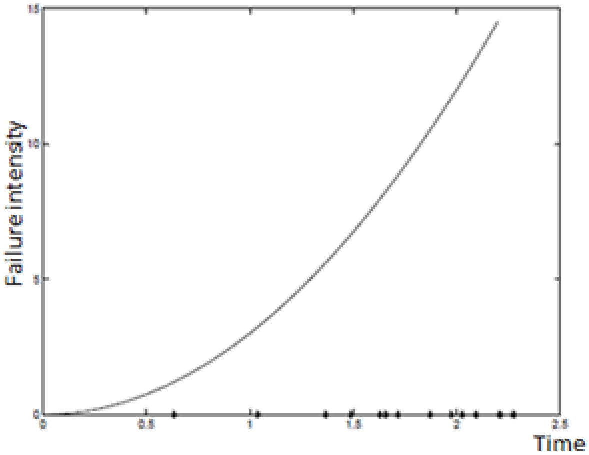

Figure 6. Failure rate and failure intensity for minimal maintenance

A simple example of minimal maintenance is similar to replacing a flat tire on a car. Because the maintenance action has no effect on the reliability after the repair, the failure intensity function has the same shape as the failure rate and increases with time.

There can be several reasons why the state of a component is unintendedly BAO after maintenance; for example, the maintenance action was performed at an inappropriate time, not according to the prescribed procedure, or the component is beyond its useful life.

The assumption of minimal maintenance is reasonable when only minor components of a system are repaired.

3.2.2.3 Imperfect Maintenance

Most maintenance activities do not result in the extreme cases of perfect or minimal maintenance but rather in a complicated intermediate situation. A maintenance action that improves the condition of a component to in between GAN and BAO is known as "imperfect maintenance." This type of maintenance is more realistic, which is the result of a combination of the quality of the maintenance procedure, reliability objective, skills of the maintenance personnel, and overall maintainability of the system.

Figure 7 shows, for a wear-out failure mechanism, the effect of imperfect maintenance on the failure intensity trajectory. In this case, the maintenance action has some level of relief on the conditional probability for the next failure (i.e., failure intensity drops) but not a total relief (i.e., minimum failure intensities show an increasing trend).

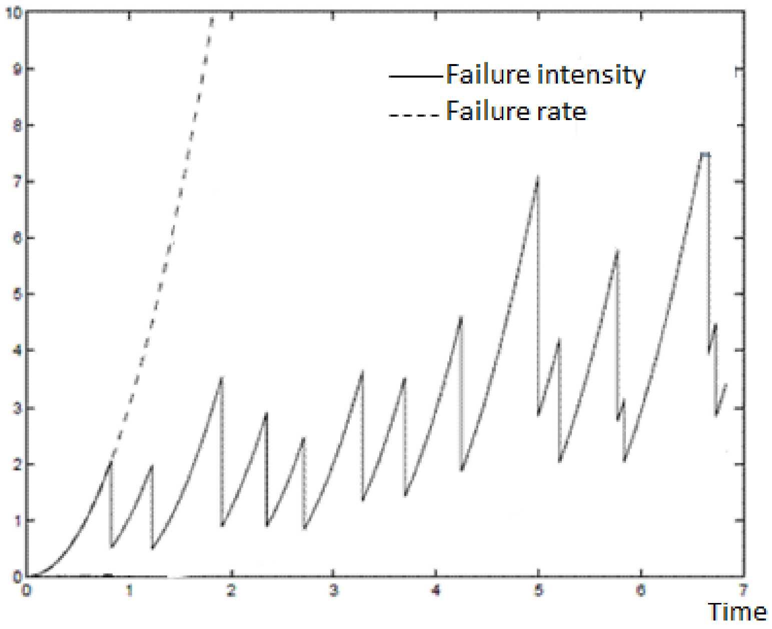

Figure 7. Failure rate and failure intensity for imperfect maintenance

3.3 CHARACTERIZATION OF FAULTS

According to the definition in the glossary (see section 8), a software fault is a manifestation of an error and occurs when the output of a function does not meet expectations. Expectations may be expressed as a required value for the function output or as performance characteristics for a required functional capability. A fault may have no impact on the user, because of designed fault-containment strategies, or may result in a failure potentially ranging from a slight visible impact to the crew without disturbance to a safety-related impact. Such impacts may be caused by the deterioration of performance at system- or aircraft-level, or by the interruption of a safety function.

With respect to fault occurrence in software, not all functions embedded in a system are active for the same amount of time during in-service operations. This is different from the software development and testing phase, in which functions are all specifically tested with normal range and abnormal ranges of input values. Software component embedding functions that may be activated only during a short period of time during in-service need to be properly tested and specified for robustness behavior; if they are not, these functions may contain dormant faults that will be activated when specific conditions occur.

The characterization of faults supposes the existence of a classification. This report uses a fault classification scheme based on the impact of the fault as captured in problem reports. The glossary (see section 8) has additional information on the lexical meaning of words used in the following sections.

3.3.1 Consequences of Faults

An EASA CM [17] categorizes open problem reports in terms of the nature and effect of the problem:

- A problem whose consequence is a failure—under certain conditions—of the system and has an impact on safety is categorized as a type 0.
- A problem whose consequence is a failure—under certain conditions—of the system and has no safety impact on either the aircraft or the engine is categorized as a type 1. Type 1 is then further divided into type 1A—if the failure has a significant[1], functional consequence—and type 1B otherwise.
- A fault that does not result in a failure (i.e., without consequence at the system functional level and not detectable by the crew in any foreseeable operating conditions) is categorized as type 2.
- Any problem that is not of type 0, 1, or 2, but is a deviation from the rules, is categorized as type 3. Type 3 is further divided into type 3A, if the deviation is significant (e.g., may lower the assurance that the software behaves as intended and has no unintended behavior), and type 3B if the deviation does not affect the assurance obtained.

[1] The meaning of "significant" is to be defined in the context of the impacted system and in agreement with the aircraft and engine manufacturers.

From the above schema, consequences of faults—whether hardware or software—can be classified according to the following:

1. Faults resulting in failure under certain conditions
    a. With safety impact
    b. Without safety impact
        i. With significant functional consequences
        ii. With no significant functional consequences
2. Faults not resulting in failure
3. Deviation from the rules
    a. Significant deviation affecting assurance
    b. Not a significant deviation

The EASA's approach to classification of effects of faults is comparable to RTCA DO-248C and RTCA DO-254 definitions in its differentiation of "fault" and "failure" (i.e., not all faults result in a failure as a fault may remain hidden). From the definition of type 2, failure can be construed as having an impact at system functional level and be detectable by the crew. The latter is not captured in the standards' definition of a failure being the inability of a system or system component to perform a required function within specified limits. This clarification is highlighted in the glossary (see section 8).

3.3.2 Root Causes of Faults

Faults observed during development tests, endurance tests, or in-service experience may have several types of causes, for example:

- A hardware physical failure, single event upset, or a system incorrect interface that will prevent a software function from sending correct output.
- A software design error.
- A lack of system specification for a specific robustness condition to which the software gives incorrect responses.

The aim of a robust software development process, as per the RTCA DO-178C guidelines, is to ensure that software components will behave as specified, and all processes put in place for software development and verification are thoroughly followed: a software fully compliant with RTCA DO-178C behaves as per its requirements, especially in robustness modes that are also expected to be specified. However, processes compliant with RTCA DO-178C guidelines will not prevent system specification errors, design errors, or lack of physical faults containment. The consequences of these types of faults may result in an unexpected software function output or calculation error.

In most cases with service history, it may not be possible to trace the root cause from the open problem report beyond the major domains of hardware, software, or systems; though in-service experience for software may be evaluated without identifying the root cause of a fault, making the default assumption that all software faults originated from software design errors ultimately may lead to more pessimistic reliability values.

In cases in which the root causes can be determined within the hardware and software domains, the benefits pertain to achieving a better accuracy in measuring and predicting the types of faults for the in-service reliability models. As an example, determining the root cause of software's observed incorrect behavior will help sorting if the software itself is faulty or behaves according to an incorrect system requirement; only the former needs to be included in the software reliability model.

Though there are no actual guidelines for determining the root causes of problems, the following sections present suitable identifications of root causes. Two main categories are identified: 1) physical faults linked to the hardware, and 2) design faults in hardware, software, system, operational, and environmental aspects of avionics systems/aircraft. The following sections address physical faults that are reproducible; design faults that are observable and reproducible; and design faults that are observable but non-reproducible.

3.3.2.1 Physical Faults

These faults are due to hardware failures or single-event upsets. They may be sub-classified into internal physical faults, due to the hardware parts of the system in which the software component is embedded, or external physical faults, due to unavailable external inputs from an external interface to the system. The modular, complex, and highly integrated nature of avionics systems/aircraft contributes to the concatenation of failure across interfacing systems. Therefore, the fault may manifest in hardware parts of the system embedding the software, but it may have an external cause. This complicates the definition of external physical faults beyond being attributable to unavailable external inputs.

Through the application of system safety analyses, the occurrence of most physical faults and their corresponding mitigation can be predicted as part of the design process.

3.3.2.2 Reproducible Design Faults

These faults result from a design process error that was not detected by the validation and verification process. This type of fault is, by definition, a problem resulting from an incorrect development or verification process. It is the main variable factor in software reliability models and also the most difficult to address. Process improvements or additional verification processes may be applied to reduce the occurrence of these issues.

It should be noted that a design software fault should be distinguished from a software design fault and system design fault error to correctly evaluate the software reliability. A system design fault may appear when a specific unspecified condition occurs and the software behaves incorrectly, from a functional point of view, but correctly regarding its own set of requirements:

- Unspecified software behavior–A type of fault observed during the in-service use of a system that can be attributed to incorrect output of a software function from a functional standpoint (as opposed to a software process standpoint). The software implementation is, however, correct in the sense that it complies with the requirements. The unspecified behavior may be traced to the system level/characteristics in the system environments that were only partially considered or perhaps never envisioned in the development processes. For example, the origin of the fault may be an unspecified system behavior or a missing software-derived requirement that would have covered the faulty behavior. Examples of modifications to remove this type of fault include, but are not limited to:

    - Adding more system requirements and modifying the software component and life-cycle data in compliance with this system change.
    - Refining the software requirements by specifying expected correct software behavior through derived software requirements approved by the safety system team.

- Interface faults–These faults stem from the interfaces among systems and modifications of interfaced systems independently from each other. A software function may be designed for a specific range of external inputs and tested with respect to this specified range. However, an unforeseen change in an external system may alter the input range. The software function may behave incorrectly with the altered range from a system point of view, though remain correct according to its own specification. This root cause is similar to the unspecified software behavior, but the fault occurring is directly dependent on the operational environment of the software function in which it is tested and used in service across several projects (section 4.2.2.4).

3.3.2.3 Non-Reproducible Faults

These faults may occur either internal or external to the system. As per the previous analysis on physical faults, these faults should also be part of the software design—more precisely, while designing the fault monitoring system for each function. However, it may happen that such fault is not easily detectable by the user, or by the software fault log message function, because non-reproducible faults are by definition temporary faults. As such, the capture of this type of fault is problematic for determining the conditions of occurrences and reproducible conditions. This type of fault may also be difficult to contain and may not appropriately be detected and analyzed if no strong debugging means are developed as part of the system. These faults are usually detected through endurance tests at an early stage of development, their occurrences measured and given as an input of the software reliability model. Occurrences of these types of faults tend to decrease over software changes following endurance tests or in-service exposure time; they may tend to increase with system change, enhancements, and operational changes.

Non-reproducible faults will usually be taken into account in the software reliability models, because they are observable in the data collected; however, because they are not reproducible, their integration in the model is performed without any filtering for particular root cause. This may cause issues as different types of faults (e.g., hardware, single event effects [SEE], or software) may be lumped together.

3.3.3 Consideration of Software Architecture and Functions

Software architectures vary widely as a function of the systems for which they are developed, languages used to develop the software components, architecture models applied, etc. In general, software functions may be comprised of several software subfunctions designed into operational layers. The upper layer executes high-level functions and interfaces with other systems. The lowest layer, developed to communicate with the hardware components, is the hardware/software interface layer.

As an example, figure 8 shows a software architecture in which data flow and control flow are authorized between two adjacent upper layers, Function_1 and Function _2. The arrows between the upper layer, the software subfunctions "a" through "e," and the low-level software layers, SW1 through SW4, may be read as "calls a function of/exchange data with," as they symbolize the control flow and data flow among software subfunctions.

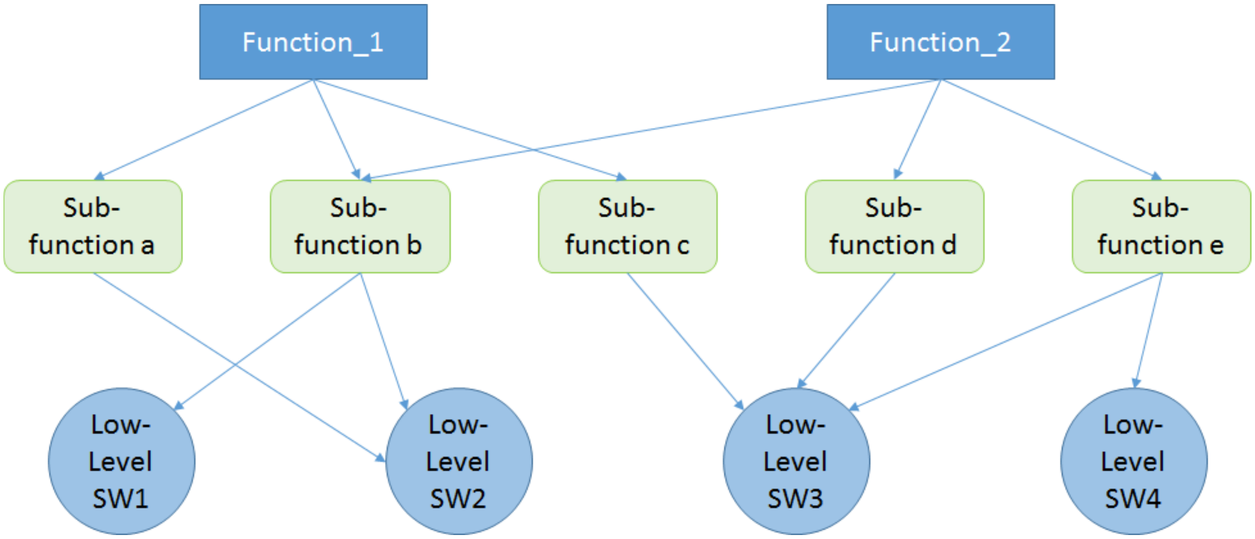

Figure 8. Example software architecture illustrating data flow and control flow between layers

This architecture gives a static view of the data flow and control flow among software functions. Additional information on the dynamic architecture may help determine the average execution time of each subfunction as a function of the possible operational states of the complete software. Though functions may not be or remain determinate, this information may be useful for debugging or as maintenance information embedded in the application; it will then need to be specified and tested.

In the example shown in figure 8, the Low-Level SW4 function may be called only in a specific degraded mode of Function_2, whereas Low-Level SW3 function is called at each minor cycle of the software scheduler and provides information (e.g., messages for Function_1 and Function_2). As a result, the number of faults per flight hour may be higher for Low-Level SW3 than for Low-Level SW4, but the fault rate of each function measured for each function's own execution time will give much more realistic results and usable reliability indicators for those software functions.

Therefore, whenever possible, software reliability measurement should be based on the fraction of time under which a definite operational state of the software is executed without any fault being raised.

3.4 SPECIFIC DIFFICULTIES WITH DATA CAPTURED IN SERVICE

This section describes the usual, universal data that are expected from the airline end user when reporting in-service events. It discusses the main difficulties.

3.4.1 Typical Data from Service Experience

In-service problem reports are built from a template provided by the product manufacturer that contains the following items to be filled in by the user:

- Event identification

  - User identification (e.g., airline company)
  - Name of the person filling in the report
  - Date of event
  - Date of report
  - Location/flight number/aircraft identification

- Product identification

  - Name of equipment/part number
  - Equipment subcomponent (if any)
  - Supplier of the product
  - Installation date of the product
  - Flight hours of this product

- Event description

    - Circumstances of the event
    - Description of observable effect
    - Download and analysis of the product failure messages (if any)
    - Impact of the event on the operating environment/crew
    - Immediate action taken into account when observing the event
    - Expected solution when such an event occurs
    - Root cause of the event (if identified)
    - Probability of occurrence of this event

- Safety impact

    - Did the event have an airworthiness impact?
    - Did the event generate a risk for the user/crew/operator?
    - Was there any damage or injuries?

Different types of reports exist, each with a specific focus and additional information requests: field return report, maintenance action report, and failure analysis report. Appendix D contains examples of such report formats.

Any additional data helpful for the event description or context may be added to the above short list, especially if the product itself is capable of logging its own activities and failure occurrences and if the user is capable of capturing such product logs.

3.4.2 Difficulties

Difficulties arise at different levels, in the collection process, and in the analysis of the event report. One of the primary sources is the difference in objectives between the various actors, namely, the airlines, repair centers, and development/production centers. The differences in objectives impact these processes:

- Investigating the safety impacts.
- Substantiating the in-service reliability of the product.
- Returning the aircraft to its operations rapidly.

These three processes have a direct impact on the qualitative aspects of the problem-reporting needed to justify the usability of the in-service data as part of a certification project (criteria under “P” in Appendix A). A major difficulty lies in establishing the safety-related impact of an event in the in-service environment as projected into the new operational environment or new configuration of the product.

The following sections illustrate the most common issues encountered with in-service data that impact the usability, relevance, and sufficiency criteria.

3.4.2.1 Missing or Insufficient Data

As a preamble, an in-service data collection is enabled by the aircraft maintenance systems. The quality and breadth of information collected depends on the specifications for these systems. Even when attention is paid in the design phase of the maintenance requirements, it may still be the case that information is missing or insufficient. Automated tools supporting the analysis of the collected information are also tributary to the quality of the onboard maintenance system design [26].

Many problems go unnoticed because insufficient information was provided. There are three common causes for missing essential data:

- Inspection or testing began before a procedure was in place to report problems or properly capture in-service data.
- When it existed, the problem-reporting form was difficult to use.
- The person who filled out the problem-reporting form was not properly trained or suffered from a lack of culture within the company to acknowledge defects.

3.4.2.2 Substantiating Reliability: Failure Determination Issue

To support the determination of in-service reliability, the number and classification of failures must be determined. This is not an easy task, with several difficulties occurring at different levels and with each of the actors:

- Airline-related

  - The operational activity (e.g., flight hours applicable to the event report) is not provided.

- Repair center/Supplier-related

  - Consider the example in which multiple components have failed within a product. Turnaround or availability requirements may lead to the replacement of a complete set of components without analysis of precisely which components failed. Therefore, the identification of the failed elements is inexact.
  - Failed components may not be preserved after part replacement; therefore, no root-cause analysis can be performed. Even when the component is preserved, analysis is not systematically performed anyway.
  - Issues related to testing means–Test benches may not have 100% test coverage. For some families of components, test means are simply not available (e.g., test program for memories).

3.4.2.3 Subjective Classification of In-Service Problem Reports

Though the classification scheme proposed in section 3.3.1 may seem simplistic, the scheme at airlines or repair centers is often times a qualitative scale distinguishing minor from major problems. The impact of problem reports raised during the in-service product life may change from minor to major depending on the changes from the in-service operational environment to the target environment, or on the product configuration. The analysis of problem reports by qualified people may be altered by the lack of a precise definition of the in-service environment or insufficient qualitative data on the product itself; this potential alteration impairs the establishment of a correct justification for these problem reports in the target environment.

4. FINDINGS FOR ACCEPTING DATA: QUALITATIVE CRITERIA

The findings in this section pertain to the bottom-up approach. They are both in support of an applicant's decision to use service history or product experience to claim certification credit and for an authority's evaluation of the applicant's claim. First, a caveat to the scope limitation of this section relates to the fact that current industrial practice of the researchers limit the consideration of service history or product experience to claim certification credit to hardware items of development assurance level no higher than D. Therefore, some of the criteria may be applicable solely to hardware, and the approach can be disregarded if decision flow is applied to the DAL higher than D. Section 6 proposes an alternative approach suitable for all DALs.

Qualitative criteria must be accompanied by the quantitative information presented in section 5.

Qualitative product service experience data refers to the extent and conditions to which the product was used while accumulating service experience. This includes used and unused functions; configurations or operating modes of the product, defects, and problems encountered; and actual conditions of its integration, installation, or environment. In addition, available data on the product itself—whether it is a software or hardware component, or an LRU—are to be taken into account while collecting the product service experience. Installation manuals, user guides, development life-cycle data, and verification reports of the product may help in understanding how the product behaves and failure monitoring is handled while in service. Based on the qualitative descriptions for the above conditions, the need for specific associated quantitative data can be defined.

Qualitative aspects discussed in the sections that follow include intrinsic properties (e.g., criticality, level of innovation complexity, and impact on end user), operating environment, processes from design to installation, and remaining open problem reports.

4.1 INTRINSIC QUALITATIVE ASPECTS

Depending on the intrinsic characteristics of the product and its in-service operational environment, one or more objectives of the RTCA DO-254 or RTCA DO-178C documents may be covered or equivalent safety level achieved by determining a correct set of relevant quantitative data.

4.1.1 Criticality

Criticality can be expressed by the associated failure condition or assurance level. The higher the criticality, the higher the acceptable level for relevance and sufficiency (including criteria for validity) would be on the product service experience to substantiate the design/development assurance and obtain certification credit. Such requirements should appear in the form of a set of relationships between the DAL or target level of safety and the product service experience, both qualitatively and quantitatively.

For software aspects, the RTCA DO-178C document is built around a set of objectives that must be achieved for a designated development assurance level derived from the system safety analysis. For a software product of highest assurance level, or DAL A, all RTCA DO-178C objectives are applicable. Conversely, the objectives associated with DAL D are limited to the development and verification of the software-component requirements with respect to the system requirements and architecture development. Therefore, identifying objectives that may be covered by software service history in lieu of RTCA DO-178C recommended activities may become a tedious task for some objectives and nearly impossible for others (e.g., objectives on modified conditions/decision coverage structural test coverage).

The use of service experience as an alternative method to achieve some RTCA DO-178C objectives has to be precisely identified and justified on a case-by-case basis; there is currently no defined rule to establish a link between development assurance level and product service history for software components. A question-based approach is proposed to support the decision of pursuing service-experience-based credit for a product based on criticality:

- What is the development assurance level allocated to the product?
- Does this product contribute to one or several safety-critical functions?
- Are the safety-critical functions already documented by the product developer?
- Has the product been developed and used in aeronautics or another domain?
- What are the objectives from current standards applicable to this product?
- What is the missing evidence with respect to the hardware/software assurance standard's objectives?
- May service history data compensate the lack of evidence on the software component for some of the required RTCA DO-178C objectives?

4.1.2 Level of Innovation

A brand new product (e.g., newly developed hardware item or previously developed hardware item used in a particularly novel application) will not offer sufficient product experience data to substantiate the design assurance for the new usage conditions. However, a product is rarely entirely new and may be derived from other products or include parts from other products that offer enough service experience.

As far as innovation is concerned, the key point is to address the certification process based not only on the development activities at hand but also to prepare for the in-service activities. It is paramount to account throughout the design for the definition and accessibility of the data that will be collected not only for in-service reliability assessment but also for the potential of experience-based certification credit (e.g., reuse). When it is not taken into account or anticipated in the design, incomplete or missing information from the data collection process is highly likely to result.

4.1.3 Complexity

The more complex a system, the more likely functional and dysfunctional modes may provide sources of hidden defects. Consequently, more product service experience, both qualitatively and quantitatively, would be required for complex products to meet the acceptability level for such data.

Though complexity may not be precisely and universally defined for all types of airborne equipment, several factors may contribute to the definition of a software/AEH complexity, including, but not limited to:

- Number of functionalities performed by the product and control coupling (e.g., nested calls and conditional structures that activate/inhibit functions).
- Complexity of implemented algorithms (e.g., Kalman filter comprising 80 states configured according to flight phases).
- Number of transitions in state machines embedded in the product (including several operational modes, complex conditions for multiple state transitions, etc.).
- Software control flow (e.g., number of nested calls and conditional structures).
- Software data flow (between external software/hardware/system components or between internal components of the software itself).
- Product configurability (e.g., deactivation of embedded functions, configuration of filters, and algorithms).
- Number and type of possible inputs/outputs of the product.

All of the above items have a direct impact on the extent of the testing activity on the product. Complexity in external inputs such as configuration, input/output, and pin programming may result in such a large number of possible configurations that their complete scope may not be physically testable. In those cases, a functional and abnormal subset of test case procedures is developed that is adapted to the context of the product.

According to a March 2012 EASA report [16], for a simple AEH component, the ability exists to demonstrate the expected operation of the device under all possible combinations, permutations, and concurrence of conditions of the inputs of the individual logical components within the device. No such definition exists for software components in the current industry standards, but the ability to test all possible configurations and corner cases may be evaluated through predefined criteria, such as the ones given as examples in the above list.

The analysis of the complexity criteria may not always be possible, because the items indicated in the bullet list imply an access to the life-cycle data of the product; however, when it is possible, this analysis has a direct impact on the appropriate service experience time exposure, as a relevant number of executions for corner-case behaviors and modes needs to be captured. The complexity criteria are also determinant in identifying the operational environment suitable for service experience-based credit. A question-based approach, as outlined below, is proposed to support the decision of pursuing service experience-based credit for a product based on complexity:

- Are there any available product design or requirement artifacts from which the product architecture and complexity may be evaluated?
- Is the information in the product documentation sufficiently detailed to establish the configuration and interfaces with the product?
- Are the capabilities, functions, modes (e.g., normal, degrading, and failure modes) of the product sufficiently documented to be correlated with the service history data?

4.1.4 Impact on the User

Though not always classified as safety critical, a hardware or software product or component may have an impact on the user (e.g., operator, crew, maintenance personnel, or a customer of airline companies) by altering either the availability or the continuity of the product function. The product may fail to perform its function, but another part of the system (or an external system) may compensate for the failure either by returning this function to a nominal mode for this user or by enabling a degraded mode. Moreover, a failure in assuring the availability of a function activated by the user may greatly alter the confidence of the customer in the software or hardware product.

Though seeking certification credit based on service experience for such products, the analysis must be performed on the continuity and availability of its functions throughout its service hours and on the system architecture in which this product will be installed so that, for example, backup modes and primary/secondary modes can be included in the analysis.

4.2 QUALITATIVE ASPECTS RELATED TO THE ENVIRONMENT

According to the existing guidance, suitability is conditioned on the demonstration that an acceptable similarity exists between the target environment and the environment that was originally used for the data collection. It is therefore necessary to define the various elements that may have an impact on the operating behavior and safety assurance for the product.

The environment configuration that must be taken into account has multiple facets that may depend on different factors. These factors include, for example, connections with the outside of the product; resources; modes or functions activated (or not) in the product; and the range of variation of the data exchanged within or outside the product. All of these elements must be evaluated both within the target configuration and within the configuration in which the data were collected to demonstrate the relevance of the results for the target configuration.

In short, the concept of "similarity" and how its assessment is performed is based on a documented comparison of a selected set of key characteristics between the original and the new contexts. The following sections detail these characteristics and highlight issues that may negatively impact the assessment of similarity. However, the quantitative criteria under which the similarity assessment reaches satisfactory acceptance remain undefined, according to the current knowledge base of the researchers.

4.2.1 Facets of Operating Environment

The first difficulty lies in the variety of relevant acceptations for "operating environment." It may refer to the actual environmental conditions in which the product performs its intended functions—for example, all conditions per RTCA DO-160G, "Environmental Conditions and Test Procedures for Airborne Equipment" [27], such as electrical, mechanical, electromagnetic compatibility, climatic, and indirect lightning. This environment may be an avionics environment (i.e., the product is embedded in an aircraft) or a non-avionics environment (e.g., ATM, railway, automotive, or nuclear). Environmental conditions in which data are collected will therefore have to be identified for the purpose of assessing the similarity with the targeted environmental conditions.

Operating environment may also point to the usage domain in which the product is operated. The concept of usage domain comprises all interactions the product has with its external world. It includes the concept of operation (e.g., operational or functional modes; normal or alternate modes; degraded or backup modes; and interaction with people or other systems) and the concept of the closed environment of the product (e.g., the physical interfaces with other systems/people and the possible configuration of the product). The concept of usage domain also encompasses the limitations within which the product can be operated in its full capacity in terms of performance and safety.

The 14 CFR 25 for transport-category airplanes, Subpart F Equipment Section 25.1309, item (e) [21], uses the terminology of "foreseeable environmental conditions." Furthermore, the FAA's AC 25-11A, "Electronic Flight Deck Displays" [28], expresses the definition as "foreseeable conditions means the full environment in which the display or the display system is assumed to operate, given its intended function. This includes operating in normal, non-normal, and emergency conditions." This definition suggests that, even within a well-defined set of usage domain and environmental conditions, the range of operating situations may vary significantly, even when taking into account only those that are foreseeable.

Consequently, the relevance of data for product service experience should be evaluated through a multidimensional structured analysis: first through the similarities between original and target usage domains (i.e., relating to functional environment), then with respect to environmental conditions (i.e., subject to installation constraints), and all the while considering other potential situations (i.e., robustness to singular/extreme cases).

Any domain may be relevant for service history as soon as similarity is demonstrated with the target domain. For example, data may be collected during in-service use of the product by airline companies, through extensive endurance tests in representative configurations of the target domain, and/or during in-service use of non-avionics domains. The challenge is to select which operating environment to use as product service experience to claim certification credit. The availability of several possibilities raises the question of whether such environmental conditions should be defined as an encompassing envelope or as a combination of several discrete environments.

4.2.2 Assessment of Usage Domain Characteristics

The characteristics of the operating environment as it relates to usage domain should be evaluated using the following key considerations:

- Physical interfaces, which include the interfaces and relationships among the internal components of a hardware or software item.
- Functional configuration.
- Exchanged data with its external world.

The objective of the analysis is to capture any event or input that impacts the product's functionalities and behaviors. Using the three key elements, possible inputs are evaluated through the physical configuration of the product (i.e., interfaces used in the original environment and in the targeted environment). The analysis includes the detection of potential issues related to the modification of interfaces or connections; the impact of changes in the functional configuration, whether performed by an actual configuration file or by pin programming; changes in workload or frequency; and differences in the type of data exchanged.

4.2.2.1 Product Identification and Configuration

The first step is to identify the product for which the activity of data collection has been performed and the gap between this original product and the target product that will be embedded in the aircraft.

Identical part numbers between the target item and the evaluated item are not systematically necessary if it is possible to demonstrate that the differences have no impact on the relevance of the collected data. Moreover, it is possible that the part number of the target item is unknown at the time of contemplating the use of service history; rather, the key is the circumstances of a required change in part number. A typical example for a hardware circuit board is a change in a component's reference that will generate a part number change in the target product. It can be easily demonstrated that a change of components, such as resistors, does not significantly alter the product's functionality. Though this step is necessary, it may not be a sufficient condition. For example, the demonstration that the product remains F^3 (form, fit, function) is not sufficient to justify the validity of collected data for product service experience.

Other examples may be found for software components for which regular fixes are performed throughout the product life; there may be software code cleanup activities that do not alter the intended functionality, a robustness fix that improves the function stability, etc. In these cases, the data collected in-service remain relevant.

In all cases of part number modification between the original product and the target product, the analysis is based on a robust product change management, including a detailed change history allowing evaluation of the functional impact of changes on the product through its service life. The "Change Impact Analysis" that is performed whenever a change is introduced should also include considerations on service history and, in particular, whether or not the collected data remains valid or is impacted by the change.

4.2.2.2 Configuration of Product External Interfaces

The external demands on the product are linked to the configuration of the external links between the product and its external world in its original domain and in the target domain. External interfaces, for example, include data links, discrete hardware pins, and power supply inputs for LRUs or hardware components. For software components, though no physical links may be explicitly named, external interfaces include any means of data exchange external to the software component or library proposed for the service history demonstration.

The key characteristics to be analyzed relate to the product means for acquiring inputs and providing outputs to external systems or people. The two main conclusions to be reached for determining the relevance of data collected during service history are:

- Whether or not the external interfaces configuration is identical between the original domain and the target domain.
- If not identical, the functional behavior has to be evaluated in both physical configurations either by analyzing the product architecture (i.e., the functional link associated to each external interface) or by evaluating both configurations in a representative endurance test environment to show evidence of the functional similarity between the physical configurations and, therefore, applicability of the in-service data in the target environment.

To illustrate this discussion, consider the connection capability of the LRU or hardware items shown in figures 9–11.

In figure 9, the product has a capacity of five external input types, among which two are redundant inputs (type 1 and type 1R). Input connections in the figure may be, for example, discrete hardware pins, analog inputs, serial links, ARINC links, or controller-area-network buses.

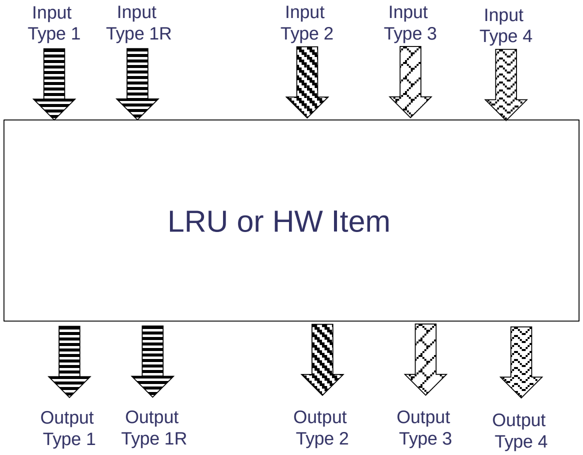

Figure 9. Example of connected LRU or hardware item

Figure 10 shows the product connection configuration in the target environment; the arrows symbolize the physical connections of the product in the target aircraft.

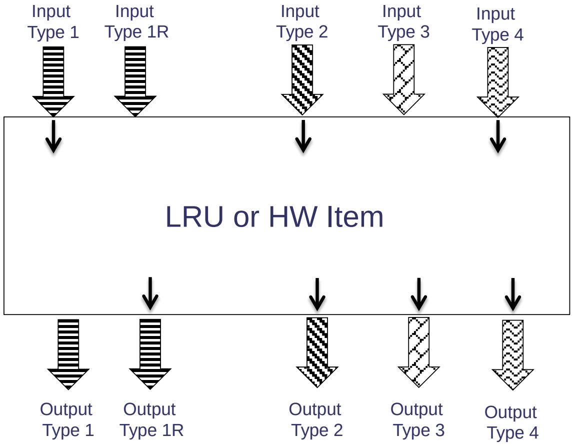

Figure 10. Example of connected LRU or hardware item in target environment

In this case, the ideal evaluation configuration used for collecting product service experience data would be this physical configuration. It would ensure that the same external interfaces have been used as in the previous environment. Any event or problem observed previously on these external links becomes relevant for the target environment.

Figure 11 shows a typical case of difference of external links between the original and target environment. The input type 1 is used in the target configuration, whereas the input type 1R was used in the in-service configuration.

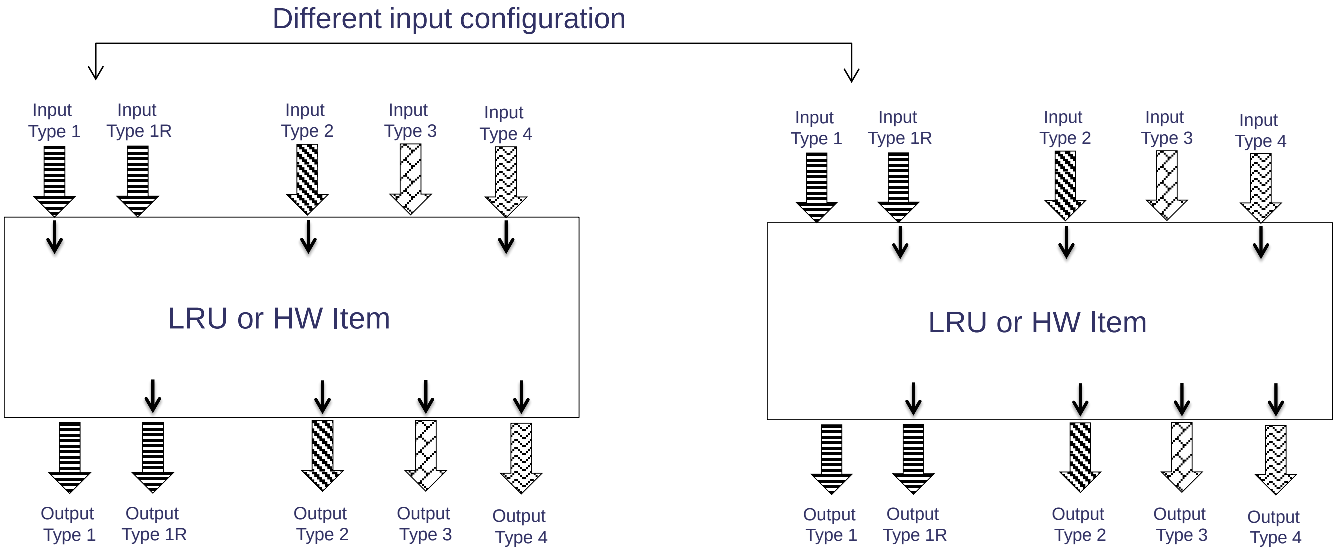

Figure 11. Identification of differences in external interfaces

Though input types 1 and 1R are redundant, an analysis should be conducted in this case using either the product architecture analysis, if available, or by testing the product in both configurations to ensure that interchanging the redundant inputs does not affect the product behavior in normal mode or degraded mode. The redundant input may, for example, trigger a different set of software functions than the other input.

The same analysis process may be applied for the output connection configuration, based on the availability of the design data; the use of a different output may lead to the activation of different parts of the software or hardware components involved in the input/output management.

4.2.2.3 Product Functional Configuration

This section addresses the internal configuration of the product that may be used to activate/deactivate hardware or software functions; configuring filters or other mathematical algorithms; configuring the hardware capacities and low-level software drivers; etc. The product functional configuration has a direct impact on its operating modes, normal modes, and abnormal modes.

The configuration of a product varies according to the type of product (e.g., hardware pin programming, configuration files containing Booleans used to disable a function, or configuration script that configures a message output for the maintenance operator). The type of configuration is not discussed in this report; however, the knowledge of the impact from possible configurations on the product functions is a key element for determining the relevance of service history or product experience collected data.

For any product, the functional perimeter may be determined by the product’s functional configuration, as described either in the product user or installation manual; in its architecture and design documents; or in both. For COTS products (avionics or non-avionics), only the user guide or installation manual is usually available; it indicates the possible configurations of the product from a user standpoint. The main issue in the functional configuration analysis between the service history original environment and the target environment is to not only evaluate the impact of activating/deactivating a function, but also to estimate the impact of changes in numerical value (e.g., filters threshold values and displays graphical objects configuration).

To illustrate this discussion, consider the functional configuration of an LRU or hardware item in figures 12 and 13. The functional configuration shown in Figure 12 may be performed in several ways:

- Parameters or resources activation/deactivation are performed through configuration data that belong to the product. In this case, the modification of this configuration changes the product part number, and the analysis for configuration equivalence is merged with the analysis of the part number change, which is presented in section 4.2.2.1.
- Parameters or resources activation/deactivation are performed through configuration data external to the product. In this case, the part number of the product remains unchanged, even if its configuration of use is modified. This may be the case of a software library that uses, as input, an external configuration file that has its own part number.

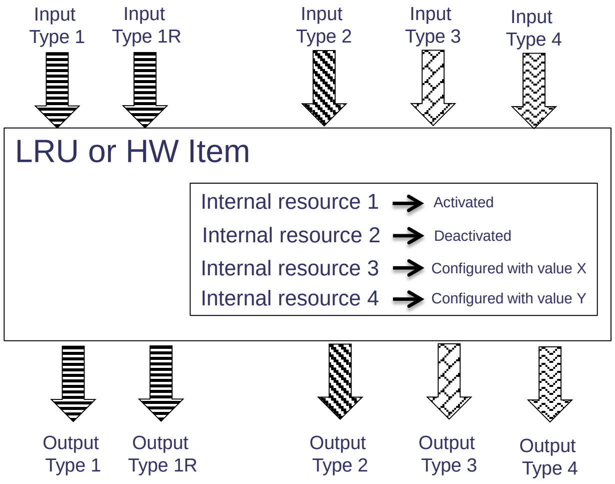

Figure 12. Example of functional configuration for an LRU or hardware item

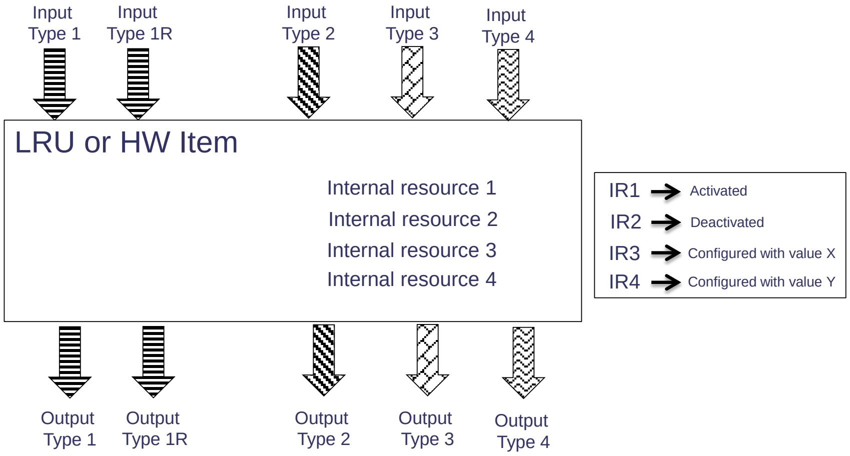

Figure 13. Example of functional configuration differences in target environment

If the target functional configuration of the product is different from the original functional configuration, the following cases should be envisioned:

- Additional information based on the design, user guide, or installation manual are available and allow for the demonstration of similar behavior between the original and target environment using additional verification, inspection, or analysis activities. In this case, service history or product experience data may be taken into consideration for certification credit.
- The analysis presented above may also be used to identify functions or modes that are activated by configuration in the target environment and that were also activated in the original environment, rendering service history data relevant for those activated functions in both environments; that is, if means for identifying product functions in the service history data collection process are available.
- No additional information is available that could support the demonstration of the similarity of behaviors between the two configurations. In this case, the service history or product experience data may not be considered for certification credit, as unknown configuration data may alter unknown functions or capacities. In a practical case, the service history data will be replaced by a new extensive test campaign to execute the product possible functional configuration for the target environment.

If the complexity of the system is such that it exceeds the capability of testing, an alternative would be to combine the test campaign with model-based system engineering (MBSE) analyses, where relevant.

4.2.2.4 Characterization of Exchanged Data

The characteristics of the data received by the product are also to be considered as a key element for the relevance of the service history or product experience collected data. Assuming that all previous criteria show that product identification, external/physical/logical interfaces, and functional configuration are representative between the original and target environment, the type of data exchanged between the product and its external world may considerably alter its functionality.

A first typical use case is the use of a serial link through which a set of messages is received by the product in the original environment. The same product may be used in the target environment to receive a different set of messages (e.g., different identifiers and/or different sizes) at a different frequency rate. In this target environment, there is difficulty evaluating how the product will behave in-service with messages size and rates that are more stringent than in the original environment—rendering the service history data collected irrelevant.

An analysis must be performed to ensure that target and original environments are similar in terms of data characteristics. The minimal key characteristics that must be taken into account for the incoming data are:

- The volume of received data, which may impact the capacity and performances of the memory in the data reception and decoding processes.
- The frequency at which the input data are received, which may also have consequences on the product performances and buffering capacities such as handling the input flow in acceptable time frames and possible loss of messages if the product only samples incoming data at a lower rate than the emitted one.
- The variation range of the data (including possible sets of message identifiers and frames), which may activate some previously unused robustness functions, in case of, for example, out of range values, counters overflows, or floating-point values cancellation effect.

4.2.2.4.1 Analysis of Data Frequency Impact

Figure 14 shows the impact of data frequency changes. In the in-service environment (top row), a software component performs an acquisition every 20 ms and declares a data-capturing failure if the emitter of the data sends its messages in a time frame exceeding 100 ms. The original emitter transmits messages at a rate of 40 ms, including a tolerance of a few milliseconds. The couple emitter frequency and receiver frequency has to be considered to evaluate whether or not the collected information may help demonstrate the good behavior of a product in the original environment.

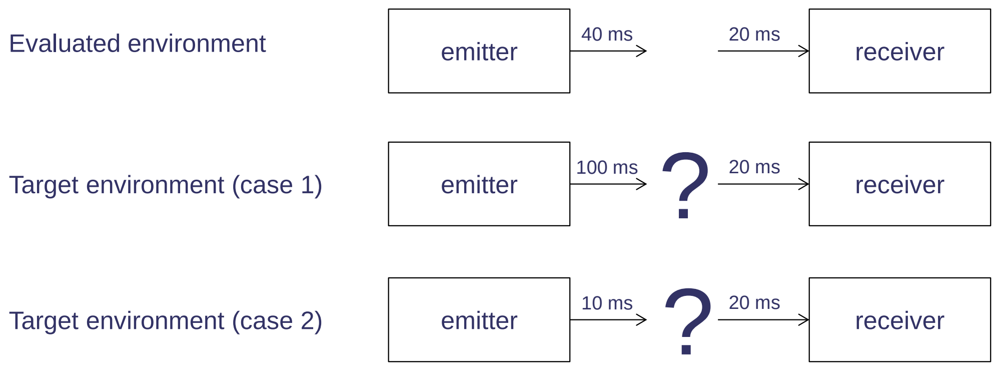

Figure 14. Illustration of data exchange characteristics differences based on frequency

In the target environment, the receiver performance remains unchanged, but the emitter frequency is modified. In one case (case 1), the target emitter transmits its messages at a rate of 100 ms; in another (case 2), it transmits its messages at a rate of 10 ms without including any potential tolerance and jitter to these rates. The changes may introduce a chance for a message to be received late and a potential failure to be consequently raised by the product due to either the message emission jittering or the sampling rate change between the product and its emitter in the target environment.

For cases 1 and 2, an evaluation (illustrated by the "?") has to be performed to demonstrate whether or not the frequency couple can be considered as acceptable in view of the fact that the collected data have been performing in a different context.

4.2.2.4.2 Analysis of Data Range Impact

With respect to data variation range, externally received data are generally contained within specific ranges that define the variation capability of the data values. Figure 15 shows the potential relations between the original range $\left[R_1^o, R_2^o\right]$ in the top row and the ranges in two cases of target environments: $\left[R_1^1, R_2^1\right]$ and $\left[R_1^2, R_2^2\right]$.

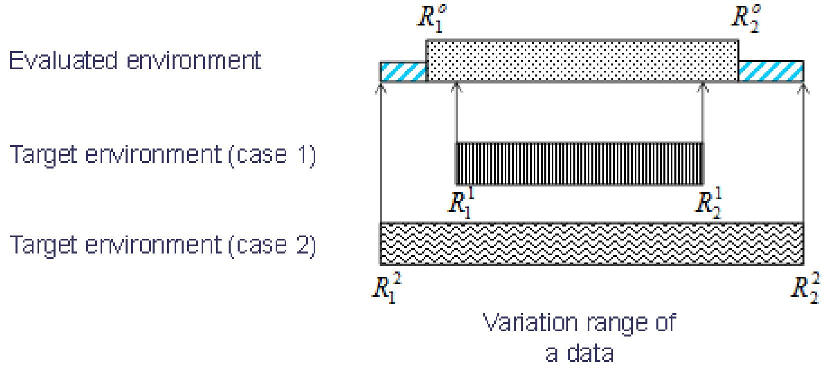
R_1^o
R_2^o
Evaluated environment
Target environment (case 1)
R_1^1
R_2^1
Target environment (case 2)
R_1^2
R_2^2
Variation range of a data

Figure 15. Illustration of data exchange characteristics differences based on range

To be considered acceptable, it should be demonstrated that the variation range of a given received data in the original environment is coherent with the variation range of the received data in the target environment. Coherence, or compatibility, is based on the fact that the variation range of the received data in the original environment is greater or equal to the variation range in the target environment (case 1). If the variation range in the evaluated environment is lower than in the target environment, additional checks must be performed to cover the gap of range (case 2) if the test cases, procedures, and results of the product are not available; otherwise, an analysis of the test procedures should show that the product is robust to the new target ranges.

4.2.3 Environmental Conditions

In the avionics domain, the correct functional behavior of the product has to be tested for the environmental conditions in the target environment. Regardless of this assessment, an analysis of the differences in environmental conditions with the original environment is needed, because these differences impact key elements in determining the validity of the product service experience in claiming certification credit. For example, if the environmental conditions were more stringent in the original environment, the data collected may contain more physical faults than what is estimated in the new environment. Conversely, if the new environmental conditions are more stringent, the increased susceptibility of the product may result in an increased number of observed faults, which should be verified by the required testing in the target environment.

As mentioned in section 4.2.2.1, a hardware change impact analysis is generally performed and documented to address both the impact of potential changes on the product and the changes in the new environmental conditions versus the original ones. The analysis aims at determining whether or not the product, as previously qualified to such environmental conditions, will fit the new conditions of the new usage (new installation).

In general, environmental conditions are established for categories and levels of severity. For example, in hardware electronics, a higher value for operating high temperature is deemed more severe than a lower one. Conversely, a lower value for operating low temperature is deemed more severe than a higher one. These few cases are easy to analyze in comparison to other cases in which the electronics are subjected to intrusive and stressing environmental conditions (e.g., lightning, high-intensity radiated fields, and electrical power interrupts). In such cases, the analysis may be more difficult and the decision regarding the validity of the product service experience more uncertain in reaching a definitive conclusion.

Moreover, as a result of the change impact analysis, new tests may need to be conducted on the product using a selected set of environmental conditions.

One approach to overcome the issue of more severe environmental conditions would be to originally qualify a hardware item to the most severe environmental conditions that are expected in-service, particularly if reuse of the hardware is intended on multiple installations. The drawback to this approach is that “one-size-fits-all” is not always feasible within time schedules, cost figures, or even meeting customer’s needs.

Units of equipment with Technical Standard Order (TSO) seem to escape such constraints, because the environmental conditions are standardized for the unit subject to a particular TSO.

From a practical standpoint, it only pushes the ultimate responsibility onto the installer of the product, leaving them to determine if the current conditions to which the product was previously qualified match their own installation needs.

Finally, as far as environmental conditions are concerned, another approach would be to design products to meet multiple categories of environmental conditions, possibly using modular design or adaptable enclosures hosting core electronics. The different enclosures would be a fit for different environmental conditions. Consequently, the product service experience for a particular module or for the core electronics could remain valid throughout various installations with different environmental conditions.

4.3 CONSIDERING AVAILABLE DATA FROM PROCESSES

The suitability of service history or product experience may also be adjusted based on the availability of data associated with the design/development process (e.g., documentation for product requirements, design, architecture, and verification reports), manufacturing and component processes (hardware items only), and installation process (e.g., users/operators manuals and acceptance test reports).

4.3.1 Design/Development Process

The following elements can be used to claim relevant experience or reinforce confidence in the similarity between the original and the target equipment:

- Derating analysis and worst case analysis contribute to a more accurate description of usage domain mainly based on tests, analysis, or simulation (e.g., thermal, electronic, or mechanical).
- Robustness or reliability tests emphasize margins over required properties.
- Endurance testing may also improve the in-service measured data through specific test conditions not easily controlled in a real usage environment.

Following up on the discussion of product complexity in section 4.1.3, available design or development data may also be used to build the product service history credit. The following data may help in consolidating the maturity of the product and the suitability of its functionality in the new context for which it is reused:

- Functional requirements
- Design and architecture documents
- Product configuration and configuration means description (e.g., software configuration files or hardware pin programing)
- Product validation and verification reports
- Any previous certification documentation

Note that available documentation needs to be shown to be under configuration management along with the product so that credit can be claimed for it.

4.3.2 Manufacturing Process

The following elements can contribute to the substantiation of product experience but are only applicable to hardware items:

- Manufacturing tests (e.g., acceptance, screening, and 100% or sampled test) highlight equipment characterization level with respect to a specific use or constraint.
- Statistical process control that provides monitoring information against deviations in manufacturing process. This type of control usually monitors specific characteristics of the product and its performance stability throughout the manufacturing process.
- Manufactured quantities are directly taken into account in the flight hour quantitative parameter but can indirectly and qualitatively inform on production flows to be compared between original and target applications.

4.3.3 Component Process

The classification of the components will play a role in the assessment of hardware reliability information, which affects the relevance, sufficiency, and problem-reporting aspects of service experience. The analysis of the bill of material allows for the sorting of the components into three distinct groups:

1. Standard components
2. Life-limited components
3. Specific components

A component is denoted as “standard” when the reliability prediction can be made using standards such as MIL-HDBK-217F [29] or, more recently, by using the FIDES (“Reliability” in Latin) methodology [30]. Examples of standard components include resistor, capacitor, and integrated circuits.

A component is denoted as “life-limited” when the wear-out failure mechanisms can appear before the end of service. Because this qualification is mission-dependent, a particular analysis called “useful life justification” needs to be performed before making the decision. If the wear-out failure mechanisms can be neglected for the service duration, the component will be qualified as “standard.”

A component is denoted as “specific” when no reliability data are available or no standard model is applicable.

4.3.4 Installation Process

Descriptive data (e.g., users and operating manuals; datasheet and architecture descriptions; and installation manuals or notices) are data that may be used to provide additional confidence by showing that a product will perform as intended within a new aircraft installation and environmental conditions. Data and documentation may also be considered as having their own service experience: a property of “robustness” can be inferred from repeated usage on multiple installations. In addition, such data must be easily usable, kept up-to-date, reliable, and secured.

4.4 PRODUCT OPEN PROBLEM REPORTS AND PROBLEM-REPORTING SYSTEMS

Depending on the type of configuration control applied to the product, the associated change control and problem-reporting systems have to be evaluated before considering the product for service history.

Product changes and their impact on the product's behavior must be formally identified to allow adequate traceability between product versions and a correct functional control throughout the product life. If the initial change control system is still in use; correctly tracks problems and corrections; and allows traceability between the product baselines; credit may be taken for this existing system.

A problem-reporting system is split into two connected parts:

1. The product manufacturer's problem-reporting system covering traceability of product changes and reasons for changes.
2. The user's problem-reporting system supporting feedback from the in-service operations to the product manufacturer.

These two systems must have a consistent interface for efficient and detailed in-service reporting leading to effective correction analysis and product updates, when necessary. Though the product manufacturer's problem-reporting system performance is tied to the development/maintenance team's testing the product, the user's problem-reporting system performance is tied to the use by, and observations from, many people interfacing with the product—increasing the variability in escalating the problem context and events description. Section 3.4 provides additional specific examples of issues related to problem-reporting systems.

An unsatisfactory problem-reporting system and change control process may alter the service experience-based credit and the capability of the manufacturer to properly maintain the product and control the events generated during the service experience.

When seeking certification credit based on service experience, the questions related to the problem-reporting system should be based on the following minimal list:

- Does a problem report tracking system exist for the product?
- Does problem reports/changes history exist for the product?
- Are the effects in the problem reports classified in terms of safety impact?
- Is a change control board set up for this product?
- Is a process between users and the product manufacturer defined to feed back problem reports and fill in relevant pieces of information?

5. FINDINGS FOR ACCEPTING DATA: QUANTITATIVE CRITERIA

The findings in this section pertain to the bottom-up approach and complement the qualitative analysis described in section 4. These findings are both in support of an applicant's decision to use service history or product experience to claim certification credit and for an authority's evaluation of the applicant's claim. First, a caveat to the scope limitation of this section relates to the fact that current industrial practice of the researchers limits the consideration of service history or product experience to claim certification credit for hardware items of development assurance level no higher than D. Therefore, some of the criteria may be applicable solely to hardware and the decision flow starts by disregarding the approach if the DAL is higher than D. When the criteria or approach cannot be applied to both software and hardware, the section clearly identifies the limitation and section 6 proposes an alternative approach suitable for all DALs.

This report discusses two quantitative criteria: 1) suitable measurement units and 2) reliability. The background in statistics used in the context of reliability measurement is contained in appendix B, and the specific aspects of modeling are presented in appendix C.

5.1 QUANTITATIVE ASPECTS OF RELIABILITY MEASUREMENT

Quantitative data, either counted (discrete) or measured (continuous), are necessary to statistically define some of the qualitative properties such as sufficiency. These data are used to measure the product reliability in-service. In brief, quantitative data are used to substantiate the achievement or the compliance with a prescribed quantitative objective as part as the certification process.

5.1.1 Input Parameters for Reliability Assessment

The SAE Aerospace Recommended Practice (ARP) 4761 defines reliability as the probability that an item will perform a required function under specified conditions, without failure, for a specified period of time. This definition lists all required input criteria that are necessary to compute the reliability figure and compare it with the prescribed objective. For a quantitative assessment within the scope of this research, the main input criteria are:

- The required function (qualitative criteria; see section 4).
- The specified conditions (qualitative criteria; see section 4).
- The period of observation (quantitative criteria discussed in this section).
- The number of failures (quantitative criteria discussed in this section).

Other lower-level detail inputs can be used in the assessment of the sufficiency of the service history/product experience, such as:

- The reliability model.
- The statistical tests used for the product maturity (i.e., constant reliability function) or the software reliability growth demonstration.
- The risk level associated with the statistical tests[2].
- The minimum flight hours required to ensure a sufficiently accurate demonstration.

[2] Risk-based testing uses risk to prioritize and emphasize the appropriate tests during test execution. Because there may not be sufficient time to test all functionalities, risk-based testing will focus on testing the functionalities that have the highest impact and probability of failure. This type of testing is mainly implemented in software testing.

The period of observation needs to be expressed in a unit that is operationally relevant for the system under study. To cover aeronautical applications, the operations to be considered are both continuous and discrete/on-event. Table 2 summarizes the units explicitly mentioned in the reference standards.

Table 2. Consolidated measurement units defined in reference standards

Parameter	Unit	Domain Applicability	Reference
Number of takeoffs/landings	Cardinal	Software, aircraft, on-event	RTCADO-248C, DP #4
Flight hours	Time [h]	Software, aircraft, continuous	
Flight distance	Dist. [NM]	Software, aircraft, continuous	
Total population operating time	Time [h]	Software, ATM, continuous	
Number of queries	Cardinal	Software, ATM, on-event	
Operating hours	Time [h]	Hardware, aircraft, continuous	EASA CM-SWCEH-001
Number of execution hours	Time [h]	Hardware, aircraft, continuous	
Usage duration in years	Time [y]	Hardware, aircraft, continuous	

For software products, an important feature to capture entails the identification of which software components or libraries in the product are more or less exposed in-service. The qualitative data described in sections 3.3.3 and 4.2 should help identify deactivated functions or functions activated only under specific conditions as compared with the functions with recurrent executions in-service. For example, a typical software product embedded in an LRU may include a scheduler, a hardware support layer, and an application component that depends on ground, flight, take-off, cruise, or landing conditions. Depending on the type of service history sought for a specific part of the software product (e.g., library, operating system, or scheduler), the measured elements expressed in table 2 may be relevant only if they are split per software component, according to the duration of specific aircraft mode conditions over the complete period of observation. As discussed in section 3.4.2, obtaining this level of detail is extremely rare and, therefore, there is a definite risk to either underestimate or overestimate reliability.

With the objective of determining reliability estimates for a product fleet, the number of units in service is an additional required input parameter. There should be a minimum requirement for the number of units in-service to make a statistically relevant data set. To the knowledge of the researchers, such a minimum requirement is not currently defined. However, it could be tied to the total population operating time criteria defined in table 2.

The question of sufficiency not only rises for the period of observation but also for the number of failures. Several intrinsic qualitative criteria offer correlation with the number of failures:

- Software source line of code–A larger size of code implemented in the product may require a larger number of observed failures.
- Software or hardware criticality–The more critical the product, the higher the level of required reliability.
- Software or hardware complexity–The more complex the product, the greater the likelihood of failure modes.

The criterion of software source lines of code, however, does not offer a direct one-to-one correlation. As an example, complex functions involving configurable filters may be more error-prone, when modifying the in-service operational environment, than a deterministic function that parses a configuration file—though the latter function can be comprised of a higher number of source lines of code. This criterion is therefore not recommended as a quantitative parameter.

The criterion of criticality is further detailed as a function of EDAL in section 6. The criterion of complexity is also covered by the approach proposed in section 6.

The in-service exposure may not be enough to give evidence of sufficient coverage of the external stress conditions or to ensure that enough robustness cases have revealed abnormal failures. In these cases, there may be a need to include an additional “simulated” service experience obtained within a specific test environment to add confidence in the maturity (i.e., stability) of the product. This practice is limited to hardware items and typically implemented in accelerated life tests (see glossary). There is no specific guidance to date on the appropriate mix of real and simulated in-service experience because it would tie not only to the question of sufficiency (investigated in this report) but also to the question of level of fidelity in the simulated service experience.

5.1.2 Reliability

Reliability is a quantitative attribute measuring dependability with respect to a given continuity of service. This attribute takes on specific instances depending on whether the product it applies to is repairable or non-repairable and whether it is software or hardware.

Experience shows that reliability can usually be represented by a so-called bathtub curve, as shown in figure 16. The physical interpretation of the three distinct segments (i.e., decreasing, flat, and increasing) is different for repairable and non-repairable systems. Further details are provided in appendices B and C.

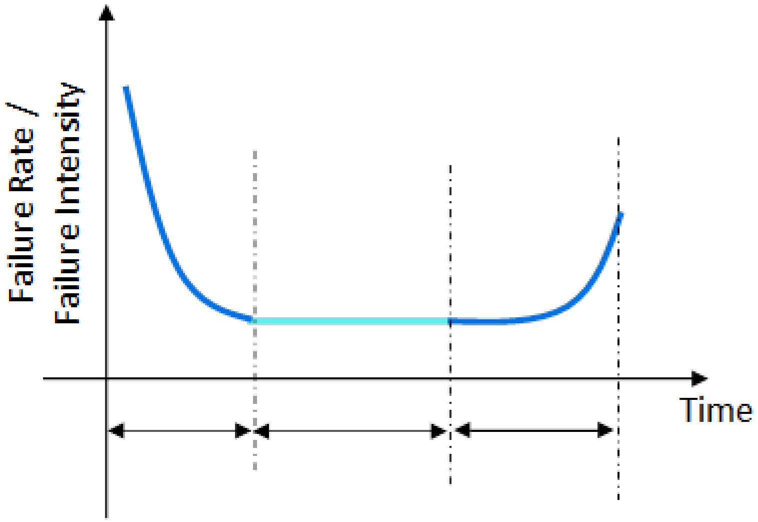

Figure 16. Bathtub curve representation

The bathtub shape of the failure rate applies to hardware physical faults and devices that wear out. For software, without corrections or modifications, the reliability is a constant for as long as the software is used. When software faults are found and "fixes" are applied (during the design phase and sometimes during the service life), software reliability is not constant but increases with the applied corrections and, therefore, with time. An analogous impact on the software reliability curve could be caused by software updates on COTS to the point at which previous versions are no longer supported, and the COTS needs to be upgraded.

5.1.2.1 Reliability Objectives

The reliability objective is usually specified by the customer, the original equipment manufacturer (OEM), at system level. However, the OEM must allocate reliability objectives at the subassembly level. Note that software can be considered a subassembly, because failure definition and mechanism are different from hardware.

For non-repairable systems, a reliability objective is typically expressed in terms of mean time to failure (MTTF), because it fits the situation in which only one failure can occur. For repairable systems, a reliability objective is typically defined in terms of MTBF. Though the two terms may be confused, the notion of using MTBF for non-repairable systems makes no sense. Note that failure in time is another way to report MTBF and is commonly used by the semiconductor industry. In the context of service history, the context for failure would be correlated to the unscheduled removal of the system/suspension of its use. Appendix C provides further details on the definition and computation of MTBF.

Though MTBF is specified as a single value, the MTBF varies over time. Therefore, different types of MTBF exist, and which type is being considered in the reliability objective should be clearly stated:

- A cumulated MTBF is computed from the initial time to the end of the observation period—for example, from entry into service to one million flight hours. This is the most used type of MTBF for in-service data.
- An instantaneous MTBF is computed from a narrow window about the observation time. It does not include accumulated information from observation period leading up to the observation time.
- A future MTBF is computed as a prediction based on the cumulated MTBF. This type of MTBF is typically used for predicting when reliability is likely to achieve its objective given a growth period. It could also be used to estimate a product end-of-life objective.

MTBF objectives are typically specified as minimum requirements, for example:

- A pitch axis rate accelerometer in a Cobham autopilot shall have an MTBF of more than 20,000 hours (MTBF objective = 20,000 hours).
- A rotor in a turn coordinator in the same Cobham autopilot shall have an MTBF of more than 8,000 hours (MTBF objective = 8,000 hours).

The following are assumptions accompanying these minimum requirements but, unfortunately, rarely elicited:

- From the customer standpoint, the mean down time is negligible with respect to the mean up time, so that MTBF is equivalent to MTTF.

- In the current guidance, only catalectic failures occur during the product life experience. Though this still might be generally true for hardware, this assumption is incorrect for software, for which the bulk of faults occurs during the development and is fixed by software corrections. For software, reliability increases with the “fixes” and, therefore, with time—which is akin to the reliability growth period in the bathtub curve.

5.1.3 Difference Between Hardware and Software Reliability

Reliability entails different elements specific to whether it is applied to an electronic hardware or a software product. Table 3 provides a non-exhaustive comparative description of reliability-related features for hardware and software.

Table 3. Comparative description of reliability features for hardware and software

In Hardware Reliability	In Software Reliability
Failures are caused by deficiencies in design, production, and maintenance.	Failures are primarily due to design faults in the software. Modifying the design can make it robust versus detectable conditions that could trigger a failure to make the repairs.
Failures are due to wear or any other energy, parts-related phenomena, but one can get a warning ahead of time.	No wear-out phenomena occur in the software; because reliability includes failures resulting in the software no longer performing its intended function, there exist hardware failures to which: 1) the software is not robust and 2) no mitigation is in place to maintain the provision of service. However, software reliability focuses on purely software errors that cause the unscheduled stop of a function. Software errors occur without previous warning. Old codes can exhibit increasing failure intensity as a function of errors induced while making upgrades.
Preventive maintenance is available and makes the system more reliable.	N/A
Reliability is time-related. Failure intensity may be decreasing, increasing, or constant with respect to operating time.	Failures occur when the logic path that contains an error is executed. Reliability growth observed as errors in the software can be detected and corrected.
Reliability is related to environmental conditions.	External environmental conditions do not affect the software reliability; whereas the internal environmental conditions affect the reliability; these internal conditions are insufficient memory and inappropriate clock speeds.
Reliability can be theoretically predicted, with some degree of confidence, from physical bases.	Knowledge of design, usage, and environmental stress factors are not factors in predicting the reliability.
Reliability can be improved by redundancy (with proper failure detection).	Reliability can be improved by software diversity; that is, making the software work with different systems [31].
Failure rates of the components in a system are predictable by analyzing the pattern of failure times.	Reliability can be improved when errors can be replicated and, therefore, corrected.
Hardware interfaces are physical connections and, therefore, can be visually inspected.	Software interfaces are not physical connections; rather, they are conceptual.
Hardware design still uses mostly standard components, though the faster-paced introduction of new technologies and shorter COTS hardware life cycles may impact this statement.	Software design does not use standard components; rather, it depends on the qualifications of a programmer, including the process being applied.

5.1.4 Software Reliability in the Design Phase

Software reliability may be expressed as a measure of the continuous availability of a specified function to the user, with the distinction of durations for which the functionality provides correct (non-erroneous) data output. With respect to the classification of fault consequences in section 3.3.1, software reliability first consists of containing faults within the software so that no user-visible failure occurs that would prevent a service from being correctly executed. Moreover, software reliability is improved when software components are modified to remove errors; the sooner errors are corrected, the lower the costs.

Software reliability is first initiated during the design phase by rigorous application of best practices and standards and, from there, building in other protections or mitigations. During the development phase, software reliability involves:

- Preventing faults by designing software components to monitor and encapsulate foreseeable faults, and developing redundancy techniques, fault containment functions, and mitigation means. This is achieved mainly by internal monitoring of safety functions or data that have been specified and tested. Error management for software components is an essential aspect of software reliability. Any possible fault and behavior related to their prevention, occurrence, and containment should be specified, designed within the software architecture, and explicitly tested through requirements-based tests in normal, degraded, or abnormal modes.
- Removing faults, through the change management process, during which:

    - Any problem reported during all development phases, reviews, tests, and in service is logged.
    - The root cause(s) of the observed problems is/are analyzed.
    - The solution for software correction is identified.
    - The impacted elements of software are modified, and verification processes are run again.
    - The modifications are tracked between software releases.

- Monitoring faults, by designing dedicated fault logs or debugging functions, to allow for investigation of the software components when faults or failures occur either during the development tests phase or during the in-service phase. These logs or debugging functions help classify in-service events and support the building of software in-service reliability models. Though not mandatory for software development or in-service problems capture and correction, software reliability statistical models help predict the in-service failure rate over time in between modifications of the software components.

The objective is to achieve the detection of errors and faults as early as possible in the development process. Methods exist, such as MBSE consistency checking; model analyses; software and model checkers; or formal methods that will support this objective. The selection of software products at the system development level that already have a satisfactory service record and existing change logs might also be considered.

5.2 RELIABILITY MODELING

With respect to the avionics domain, reliability based on observed in-service events and predictive statistical models is used throughout the embedded products' in-service life:

- To assess the product's maturity (hardware and software).
- To determine associated maintenance actions (hardware).
- To define the product's retrofit needs (hardware).
- To provide feedback for future modifications or development of new products based on technologies monitored on in-service products (hardware and software).

For the researchers' affiliation, reliability using hardware product experience and associated statistical models may be used as means of compliance for certification purposes while developing embedded aircraft products with DAL D. Reliability modeling applied to software seems more difficult to successfully use, especially in view of the lack of industry-based consensus on the deployment and use of models.

With respect to the research objectives, reliability modeling may be used to obtain values for MTBF.

Though the industry has not reached a consensus on the deployment and use of reliability models as a means of compliance, and though other methods exist to demonstrate product maturity, this research specifically entailed the investigation of models and criteria that could be used to justify results obtained by statistical modeling.

Statistical modeling of reliability is just a specific domain application of the broader field known as statistical analysis of lifetime data. The statistical descriptions and statistical properties applicable to lifetime data are provided in appendix B. The details about reliability models typically used for hardware and software reliability modeling and the computation of statistical properties and MTBF are provided in appendix C.

5.2.1 Investigating the Hazard Function

The hazard function of a lifetime distribution is a conditional probability (see appendix B) that represents the instantaneous rate of the characterizing event associated with the lifetime distribution:

- Rate of death at time, *t*, in a survivor function context for a given population.
- Rate of failure at time, *t*, in a reliability function context for a given sample of products (set of manufactured items for hardware or a software application).

The hazard function is of particular interest for several reasons:

- Because it is indicative of how the risk of the event (e.g., death or failure) varies with time, it is the target of most investigations.
- Information on its shape is a factor in the selection of the appropriate model for the lifetime data.
- It is the best suited approach for lifetime data in which the factors affecting the data vary over time.

Figure 17 shows the basic shapes for the hazard function that have been proven useful in practice [32], to include:

- Roughly constant hazard functions (a)
- Monotone increasing hazard functions (b)
- Monotone decreasing hazard functions (c)
- Bathtub-shaped, or U-shaped, hazard functions (d)
- Inverse bathtub-shaped hazard functions (e)

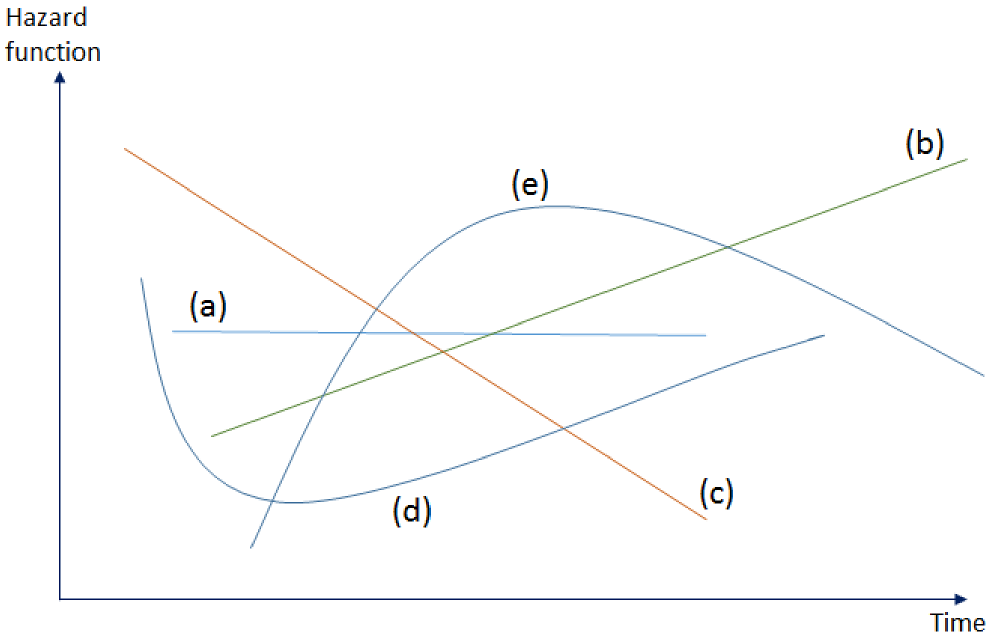

Figure 17. Basic shapes of the hazard function

The bathtub-shaped hazard function is found in most patterns of biological populations (e.g., death rate) or populations of manufactured hardware items (e.g., failure rate). Distributions with increasing hazard functions are indicative of processes such as aging or wear-out. Note that if populations exhibiting a bathtub-shaped distribution are purged, the residual population often displays an increasing hazard function. Inverse bathtub shapes are also fairly common, notably to describe the rate of survival after cancer treatment or when analyzing the duration of marriage. Decreasing hazard function is indicative of reliability growth for hardware items or typical of software reliability. Finally, roughly constant hazard functions are indicative of stable settings in which the characterizing event is due to random phenomena external to *T* (e.g., death due to an accident, hardware failure due to a shock, or software failure due to SEE).

For the avionics domain, the methods and models of lifetime data are primarily used for reliability assessment. In this application:

- The characteristic event is a failure.
- The hazard function is termed "failure rate" for non-repairable systems and "failure intensity" for repairable systems. Its mathematical notation is $\lambda(t)$.
- The survivor function is termed "reliability function." Its mathematical notation is $R(t)$.

Table 4 summarizes key reliability parameters applied to non-repairable and repairable systems (see appendix C for more details).

Table 4. Synopsis of reliability characteristics

λ(t) - Parameter	Non-Repairable System	Repairable System
Definition	$\lambda(t) = \lim\limits_{\Delta t \to 0} \frac{P_r[\mathrm{t} \le T < t + \Delta t \mid T > t]}{\Delta t}$	$\lambda(t) = \lim\limits_{\Delta t \to 0} \frac{P_r[f(t, t+\Delta t) \ge 1 \mid Ht]}{\Delta t}$
Shape [33]	Bathtub (hardware)	Bathtub (hardware) Bathtub/exponential (software)
Decreasing segment	Juvenile period (early failures)	Improvement period
Constant segment	Useful life (catalectic failures)	Maturity period (catalectic failures)
Increasing segment	Wear-out period	Deterioration period

For non-repairable systems (hardware):

- In the juvenile period, failures can be produced by:
    - Weak points or defects in the materials.
    - Imperfections pertaining to quality control.
    - Variability of processes and manufacturing tools.
    - Human error.
    - Inadequacy of material and human qualities.
    - Inappropriate solutions to technical problems.

- During useful life, failures observed in the field are mainly due to random phenomena or non-identified causes. They are perceived as sudden and may be due to:
    - Overloads (e.g., overvoltage, overpower, mechanical stresses).
    - Temperature fluctuations.
    - Human errors.
    - Accidental actions.

- The wear-out period corresponds to normal wear of a component approaching its end of life.

Usually, non-repairable electronic components are in their useful life when the product is in service. The associated catalectic failure mechanism is well modeled by the exponential distribution because it is the only probability distribution with the memoryless property (see glossary, section 8).

For repairable systems (hardware and software):

- In the improvement period, the failure intensity decreases due to repairs (hardware) or corrections (software). During the design phase and sometimes during service life, software faults are found, and corrections are performed to fix them. Such failures are considered as early failures and, in this case, software reliability increases with applied corrections and, therefore, with time. When COTS software is updated for users outside the aerospace domain, a similar impact on the reliability function may be observed.
- In the maturity period, the failure intensity is more or less constant. Failures are random in nature (for hardware and software). Without corrections or modifications, software reliability is considered constant for as long as it is used.
- In the deterioration period, the failure intensity increases again in spite of repairs. For hardware, failures are associated with wear-out. Though software aging exists (see glossary, section 8), more research into its occurrence in avionics applications is needed to conclude on the need for modeling. The current hypothesis is that software failures, in the absence of corrections or modifications, are catalectic failures.

Maintenance actions must be taken into account and the reliability of such systems is typically modeled by homogeneous Poisson processes (HPP) for both hardware and software (neglecting software aging). These stochastic processes describe the occurrence of failures and also share the memoryless property.

When modeling various layers of applications within aircraft systems, the result could be a sheaf of failure intensity curves unique to the characteristics of the different applications. Factors driving the shape could be significantly different. These include, for example, environmental factors (e.g., temperature, power levels and fluctuations); derating or uprating considerations; and varied life stages.

5.2.2 Categorizing Models

This research uses two main criteria for categorizing the models: 1) whether a model is continuous-time or discrete-time and 2) whether the model is parametric or non-parametric. The first criterion will be a choice by the modeler, because there exists ways to transfer from continuous-time to discrete-time. The second criterion will drive the formulas to be used to compute quality of fit metrics and MTBF.

5.2.2.1 Discrete-Time and Continuous-Time Reliability Models

Though software and hardware systems are different (e.g., their environmental conditions vary, their failure causes and failure consequences are dissimilar), the probabilistic definitions are identical and the theories of probability and statistics similar. Poisson processes (continuous-time models) can be used for both hardware and software reliability modeling.

Software reliability models attempted, at first, to describe the process of bug discovery in the development process using test data. Therefore, models often use data indicative of time between bug discoveries or counts of discovered bugs. The structure of the data and, by association, a reasonable model for the data, is dependent on the type of software, its use, and the circumstances under which the data are collected [34].

Generally, software reliability models are classified into two groups:

1. The first group contains models, which use machine execution time (i.e., computer processing unit time) or calendar time as a unit of the fault detection/removal period. Such models fall under the denomination of continuous time models.
2. The second group contains models that use the number of test cases as a unit of the fault detection period. Such models are discrete time models because the unit of software fault detection period is countable. A large number of models have been developed in the first group, whereas there are fewer in the second group.

There is what could be called a “convenience-based” back and forth between continuous time and discrete time at different levels. In some cases, it can be more convenient to work in continuous time because of the larger availability of models, whereas in others (such as implementing or coding), discrete time is best suited:

- Most models for reliability assume the time to be continuous, though there are natural cases where time is discrete (e.g., on demand system), and the lifetimes are expressed in terms of working periods or cycles.
- For some continuous time models, the application of a numerical method first requires a discretization of the model.

5.2.2.2 Parametric and Non-Parametric Models

Statistical models are of two types: parametric and non-parametric. A parametric model in statistics is a family of distribution functions that can be described by a finite set of parameters. Conversely, a non-parametric model represents more subtle aspects of the data. It allows more information to pass from the current set of data that is attached to the model at the current state, to be able to predict any future data. The parameters are usually said to be infinite in dimensions and, therefore, can express the characteristics in the data much better than parametric models.

Parametric models are preferred whenever possible because:

- Their parameters capture structural information of the data that can be used to explain past observations and support predictive capability.
- The computations of quality of fit metrics and the computation of MTBF may be in closed form and easily implementable in software.

Estimating the parameters in a parametric statistical model based on a given set of observed inputs and past observations is referred to as solving the inverse problem. Different methods exist to solve the inverse problem, including:

- Graphical method
- Method of moments
- Maximum likelihood estimator (MLE) method
- Least square estimator (LSE) method

The MLE method is the one most often used because it provides an appreciated insight into physical processes and may lead to close-form expressions. The graphical method is not accurate. The method of moments may lead to more complex calculations and, based on the context, may be meaningless. Finally, the quality of the LSE method varies with the collected data. More details on these methods are provided in appendix C.

When model parameters have no closed-form expression, statistical properties of model parameters can be estimated by bootstrap simulations. The terminology of “bootstrap” techniques points to methods of statistical inference dating to the end of the 1970s, which required extensive computations. The objective is to determine statistical properties of a random variable such as its estimate, dispersion (e.g., variance or standard deviation), confidence boundaries, and even hypothesis test. The techniques are based on simulations (e.g., Monte-Carlo) and Bayesian numerical methods (e.g., Gibbs sampling and Metropolis-Hastings algorithm) but do not require supplemental information on the sample (covariate). Through the technique, new samples are generated from the initial set, such that the term of “resampling” is also used in reference to the methodology.

5.2.3 Metrics for Assessing the Model’s Quality of Fit

To obtain a suitable estimated reliability value from the model, quality of fit needs to be measured between the model and the data. Regardless of the type of model (parametric or non-parametric), a good estimate should:

- Be unbiased.
- Have a small variance.
- Be efficient.
- Be consistent.

An estimator, $\hat{\beta}$, is said to be unbiased if its mean equals the target (or true) value, β. That is, an estimator is said to be unbiased when it does not systematically overestimate or underestimate the truth. The bias is defined as:

$$bias(\hat{\beta}) = E(\hat{\beta}) - \beta \tag{1}$$

A small variance, *var(* $\hat{\beta}$ *),* indicates that the estimator’s output has a small spread or variability.

An estimator, $\hat{\beta}$, is said to be efficient if its mean square error (MSE) is minimum among all other estimators. The MSE is defined as:

$$MSE(\hat{\beta}) = E\left[\hat{\beta} - \beta\right]^2 = bias^2(\hat{\beta}) + \mathrm{var}(\hat{\beta}) \tag{2}$$

When comparing two estimators, $\hat{\beta}_1$ and $\hat{\beta}_2$, the relative efficiency

$$Eff_R(\hat{\beta}_1, \hat{\beta}_2) = \frac{MSE(\hat{\beta}_2)}{MSE(\hat{\beta}_1)} \tag{3}$$

is used. If the relative efficiency is less than 1, $\hat{\beta}_2$ is more efficient than $\hat{\beta}_1$.

Finally, an estimator, $\hat{\beta}$, is said to be consistent if, as the sample size, *n*, goes to infinity, $\hat{\beta}$ statistically converges to β, i.e.:

$$\forall \varepsilon > 0, P_r\left(\left|\hat{\beta} - \beta\right| > \varepsilon\right) \to 0 \tag{4}$$

Using Chebychev's rule, this condition can be expressed using the MSE: an estimator, $\hat{\beta}$, is said to be consistent if, as the sample size, *n*, goes to infinity, $MSE(\hat{\beta})$ tends to zero.

In computing the confidence bounds for an estimator, the quantile function is another useful statistical quantity. For a given probability in the probability distribution of a random variable, the quantile function is defined as the value at which the probability of the random variable will be less than or equal to that probability. The quantile function is related to the inverse of the distribution function.

Table 5 summarizes the primary properties to be used when describing the quality of the estimator used in parametric and non-parametric approaches. The specific mathematical formulae to compute the metrics are provided in appendix C.

Table 5. Summary of quality of fit metrics

Metrics	Parametric Approach	Non-parametric Approach
Bias	X (expectation)	X (mean absolute value difference, magnitude relative error, mean magnitude of relative error, absolute residual, median of absolute residual)
Variance	X (variance)	X (magnitude of error relative to the estimate, mean magnitude of error relative to the estimate)
Efficiency	X (MSE)	X (MSE in non-parametric form)
Consistency	X (MSE)	X (MSE in non-parametric form)

5.3 APPROACH TO MODEL SELECTION

5.3.1 General Comments on Lifetime Data

Lifetime data cover data collected for situations in which the time to the occurrence of some event is of interest. The term "time response data" is also sometimes used in that context. Any data that describe a specific event over a certain duration are considered lifetime data.

When considering modeling and statistical analysis of lifetime data, several considerations play concurrently:

- The level of details needs to be commensurate with the specific objectives of the analysis.
- The background information about the covariates and distributions of the lifetime data needs to be available.
- The data need to be available both in quality and quantity to fit the models and check their adequacy.
- The software to perform the model-to-data fit, analyze the data, and provide interpretations needs to exist or be implementable at a reasonable cost.

For the domain of interest—reliability—lifetime data shall contain information on the event of interest (failure occurrence) over a duration. That duration relates to the determination of an operationally relevant measurement unit and can be, for example, operating hours for software/hardware systems or aircraft lifetime. The above bulleted items could translate into the following considerations:

- If the model is to be used for reliability estimation at an integrated circuit level, service data collected at the equipment level are not sufficiently detailed.
- Background information on the relative likelihood of various failure modes and historically fitting probability distributions for the semiconductors in a board may help decompose reliability data into finer components.
- The confidence in the reliability estimate is directly related to the quality-of-fit of the model to the data.
- The more data (in quantity but also in duration), the more behaviors can be captured (e.g., wear-out with longer durations and temperature dependency with varied environmental conditions).

As it applies to service history, additional considerations will play a role in the model selection through some of the criteria. In particular:

- The quality of fit as it depends on the quantity of data–Service history is by nature truncated and most likely partial/incomplete.
- The suitability of the model hypotheses–Maintenance and corrective actions are typically imperfect.

5.3.1.1 Discrete Versus Continuous Time Models

Most of the standard methodology for lifetime data has been developed for continuous time models and, therefore, most of the supporting software applications. It is often the case that even though time is discrete (e.g., number of cycles to failure, number of demands on an application), the model is selected among continuous time models. Conversely, it is also possible that to use a specific numerical method, the continuous-time model needs to be discretized.

5.3.1.2 Parametric Versus Non-Parametric Specifications

The selection of parametric versus non-parametric specifications can be based on several criteria, including:

- The amount and type of data available.
- Background knowledge that specific parametric forms exist.
- The assumptions on the smoothness of the underlying distributions.
- The objectives of the analysis.

In general, both parametric and non-parametric aspects are combined in the selected approach. The advantages of parametric models cover the simplicity of implementation; usability of likelihood-based inferences; and ease of use for tasks such as description, comparison, prediction, and decision-making. Conversely, non-parametric methods are less restricted by assumptions compared with parametric methods.

The selection of a specific parametric model will be substantiated by its tractability and ability to fit the data. The model first needs to capture all features of the distribution that are apparent from the empirical data; it should then be capable of representing the perceived features of the hazard function. Finally, it should adequately mold the behavior of the right or left tails of the distribution.

5.3.1.3 Model Comparison

Even in the simple case in which no covariates are present, the superiority of one model over another often needs large data samples. Therefore, the presence of right censoring limits the comparison that can be performed. This finding feeds an already strong tendency to use mathematically, or at least computationally, convenient models.

Whatever the model, it is only an approximation of reality so that:

- Observed data may be described by several models equally well.
- Observed data may contain more than one possible interpretation.
- Most conclusions or recommendations for actions are sensitive to the selection of a model.

5.3.2 Selection of Software Reliability Model

From the partial list in appendix C, numerous models allow for the evaluation of reliability (e.g., MTBF) and predicting future behavior based on the knowledge of the failures and successive corrections of software. Using these estimates, a criterion for assessing tests could be provided, maintenance could be performed, fixes could be guaranteed, etc.

The abundance of software reliability models and lack of a universal model contribute to the difficulty for users to choose the most suitable model for their problem. Relatively few studies have been devoted to the comparison and selection of software reliability models. This investigation moves one step forward by proposing a decision flow process and criteria to be used when comparing or selecting reliability models.

The selection of models takes place in three stages:

1. The first step is to determine the relevance of the models compared to the problem under investigation; that is, check the relevance or validity of the model's assumptions.
2. The second step relates to determining the intrinsic quality of a model. Is it simple? Do the parameters have a physical meaning? Can estimates be obtained easily and accurately? Does the model allow for accurately estimating the reliability?
3. Finally, the model must be closely aligned with the data. A statistical adequacy test should be performed to know whether or not the data can be fit into a model. These quality-of-fit tests typically yield good results when the observations are independent random variables and follow the same law. In fact, when observations are independent, their distributions can be numerically added; if they follow the same law, the overall shape obtained from summing the individual shapes is also representative of that law. The confrontation of a model of that law with these summed observations will therefore be good.

Five major validation criteria can be used to answer stages 1 and 2 above. They allow for the evaluation of intrinsic qualities of the models, regardless of observed data. These criteria are:

1. Validity–The model considered for selection must be based on plausible and acceptable assumptions by software engineers.
2. Applicability–The model must be able to be used in various circumstances and cases—different operational environments, different stages of the life cycle, etc. Furthermore, the model must have a certain robustness to deviations in the assumptions.
3. Capability–The model must be able to estimate, with sufficient accuracy, the attributes relevant to the users (e.g., MTBF and failure intensity).
4. Simplicity–The model must be conceptually simple, its theoretical foundations accessible to software engineers. Following the Occam's razor principle, among competing hypotheses, the one with the fewest assumptions should be selected.
5. The collection of the necessary data for the estimation of these parameters should be relatively easy and not cost-prohibitive but should integrate limitations due to intellectual property. Underlying calculations must be easily programmable and relatively inexpensive in terms of computation time.

In addition to these data-independent selection criteria, two criteria are defined that consider the data and allow measuring of:

- The replicative capacity—i.e., the ability of the model to fit observed data.
- The predictive quality—i.e., the ability of a model to predict future failure data to meet the objectives of reliability.

To support both sets of criteria, two approaches are considered; one focuses on the underlying physical phenomena, the other on favoring the model's mathematical properties. From the selection of model-fitting approaches in appendix C, MLE or LSE methods are preferred. The

LSE method is based on the method of least squares that is completely mathematical. In contrast, the MLE method is based on a physical point of view.

5.3.2.1 Physical Point of View

The type of input data, discrete time, or continuous time data may direct to the corresponding model family for selection. Another consideration relates to the fact that the main evolution of software reliability is usually performed during the development phase. That is, one usually observes a reliability growth during this phase because an intensive number of tests are performed; converesely, software maturity is observed during the exploitation phase. As reliability growth is observed, software reliability for the exploitation phase must be forecasted and monitored to discern if the reliability requirement remains fulfilled. Therefore, the selected model should support these capabilities.

It is usually possible to have an idea of the behavior of the failure intensity as a function of time. Reliability growth is often observed so that two limit values of the failure intensity need to be defined:

- The initial value $\lambda(0)$ or $\lim_{t \to 0^+} \lambda(t)$
- The final value $\lim_{t \to \infty} \lambda(t)$

The shape of the failure intensity versus time can also be a good input:

- Constant
- Monotonic decreasing function
- S-shape

5.3.2.2 Mathematical Point of View

Appendix C lists two mathematical estimators of particular interest: the MLE and LSE methods. With the LSE method, a mathematical transformation is needed to linearize the failure intensity expression, and it does not always exist.

5.3.2.2.1 Considering Input Data

Without loss of generality, a mathematical model has a certain number of parameters. One may believe more parameters are better to fit the data. This affirmation is correct if the size of input data is large. When there are only few input data, a simpler model (with fewer parameters) is often better than a more complex model because of the higher uncertainty on complex model parameters. This assertion is usually demonstrated via the use of the MSE method (even if other metrics are possible).

5.3.2.2.2 Considering Estimator Statistical Qualities

To obtain the estimator quality descriptors defined in section 5.2.3 and appendix C, two cases should be considered:

1. A closed-form expression exists–The statistical properties of the estimators can be evaluated from a general point of view.
2. No closed-form expression exists–Numerical methods must be used to estimate the model parameters. These methods are typically:

   a. Bootstrap simulations–This method can be used on a case-by-case basis because of its use of data resampling. From the original data set, input data is regenerated with the model and used to estimate the parameter.

   b. Monte-Carlo simulations–This method is applicable but a large amount of simulation is needed to find a general way of modeling the parameter statistical properties.

No model is better than all others for all cases. All of the statistical properties described in section 5.2.3 and appendix C can be used to express the difference between the estimated and parameter true value. However, because it is easier to use deterministic quantities than random quantities, the expected value is the most used property. Therefore, when in-service data can be processed with a parametric approach, the MSE method is the only possible choice. For a non-parametric approach, the MSE, mean absolute value difference, mean magnitude of relative error, mean magnitude of error relative to the estimate, or median of absolute residual can be used.

Performance objectives should be defined so that the values obtained using the above metrics can be evaluated and the selection of the model substantiated quantitatively.

5.3.3 Fit Objectives and Usability as a Function of DAL

For aeronautical software components, any means of measuring and predicting software reliability must take into account the safety parameter (i.e., the rates of safety failures that may occur must be assessed to be lower than the assigned DAL assigned by the system safety analysis).

With respect to the current literature for multiple software domains, the best results for software reliability models give a failure probability of approximately one failure every 10^6 hours. From these results, one may find elements from in-service history and failure probability that may be used to help demonstrate maturity for software components used within a system of development assurance level lower than DAL C. Software maturity could be used in the evaluation of reuse of previously developed software on a new program. Service history could be used on systems DO-178 DAL D as it allows embedding code other than the code strictly executing the intended function without having to “clean up/delete” the unused portion of the code (no test-based structural coverage). This is convenient when reusing COTS for which there is no access to the source code. This approach is no longer acceptable with DAL C and not easy to determine a service history equivalent (see open discussions in section 7.5).

Based on the classification of fault consequences and identification of root causes discussed in section 3.3, the approach to using service history for high DALs would entail refining best achievable results for software reliability models and investigating their condition of use as additional evidence to increase the level of confidence in the software maturity. This would entail the modification of the data collection and problem-reporting process to ensure that only the relevant failures are collected. Moreover, the software reliability information derived from its model could be used to substantiate the selection of the software as part of a new development effort or evolution of the existing system embedding the software.

5.4 EXAMPLES

5.4.1 Catalectic Failures Against Technology Scaling

The discussion in this section briefly highlights some of the issues arising from using reliability models to predict reliability primarily in hardware and at a component level. Further discussion on reliability at more complex/integrated levels is provided in the recommendations for future research (section 7.5.2).

One of the main hypotheses for reliability indicated throughout this document is that systems in service are mature; that is, the failures are catalectic, or the failure intensity function is in its constant portion. The entry into fatigue or wear-out conditions is typically discouraged via the integration of maintenance requirements or useful lifetime requirements (included in the MTBF objective).

Figure 18 shows a typical failure rate function observed in-service for classical technologies. The early failures are eliminated by appropriate burn-in; during in-service life, the slow growing failure rate due to aging is compensated for by the steady decrease in early failures so that the resulting failure rate curve is more or less constant.

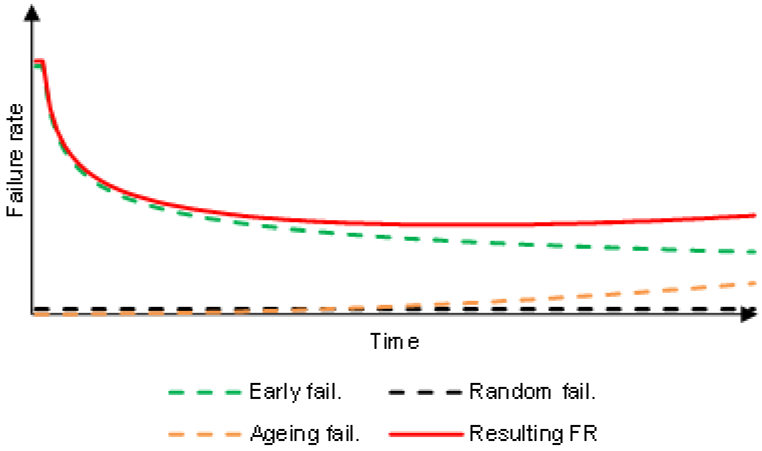

Figure 18. Typical failure rate observed on classical technologies [35]

The shape of the failure rate function is well modeled by an exponential distribution and its value is equal to the inverse of the MTBF. The estimator is simply computed from the number of grouped failures and associated functioning hours over the fleet of equipment or systems. In addition, the following is currently applied for maintenance operations [35]:

- Preventing maintenance is not needed because it has no impact on the reliability metric.
- Stock exchange for maintenance actions is easy to estimate.
- Direct maintenance costs are directly expressed from the estimated MTBF.
- The probability of failure during the "at risk time," $t_{R,}$ is independent from equipment or system age and is proportional to the failure rate, λ as $P{\sim}\lambda.t_R$.

Figure 19 shows the impact of an exponential growth of the contribution from failures due to aging onto the overall failure rate function. This increase at best reduces the constant portion of the curve but may simply eliminate it. Using a simple exponential model is no longer capturing the features of the failure rate function. Moreover, all of the above hypotheses are no longer valid, with a direct impact on maintenance actions and maintenance costs.

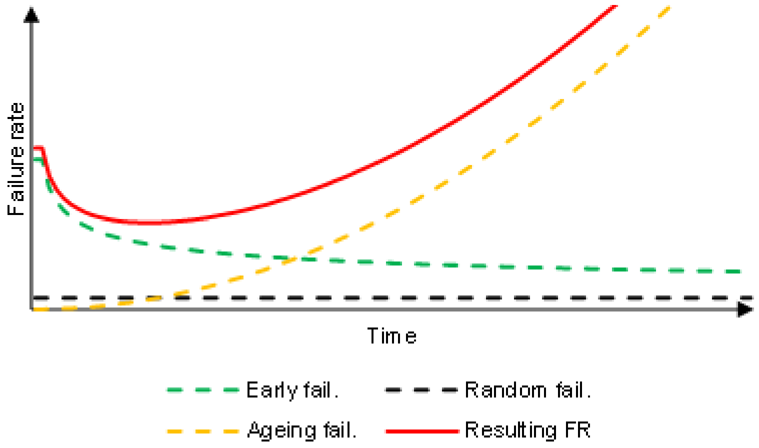

Figure 19. Issues with model validity arise when the wear-out life creeps into the in-service lifetime portion

This behavior is now being observed with deep submicron technologies. Making the issue of selecting a model even more complicated is the fact that the shape of the failure rate due to aging is dependent on the manufacturer (manufacturing processes).

At the system or equipment level, the earlier increase in failure due to aging can be mapped against lifetime requirements for avionics systems and DALs.

Figure 20 shows the evolution of the bathtub curve driven by scaling of the technology that occurs at a higher rate than reliability improvement. The result is two-dimensional: components no longer comply with required lifetimes for avionics systems (indicated as 30 years), and/or the component failure rate pushes their application to lower level DALs [36]. The premature aging increases the likelihood that the component will fail during a piece of equipment's warranty period. Finding adequate countermeasures to prolong the lifetime is to be balanced against cost, power, and/or performance penalties.

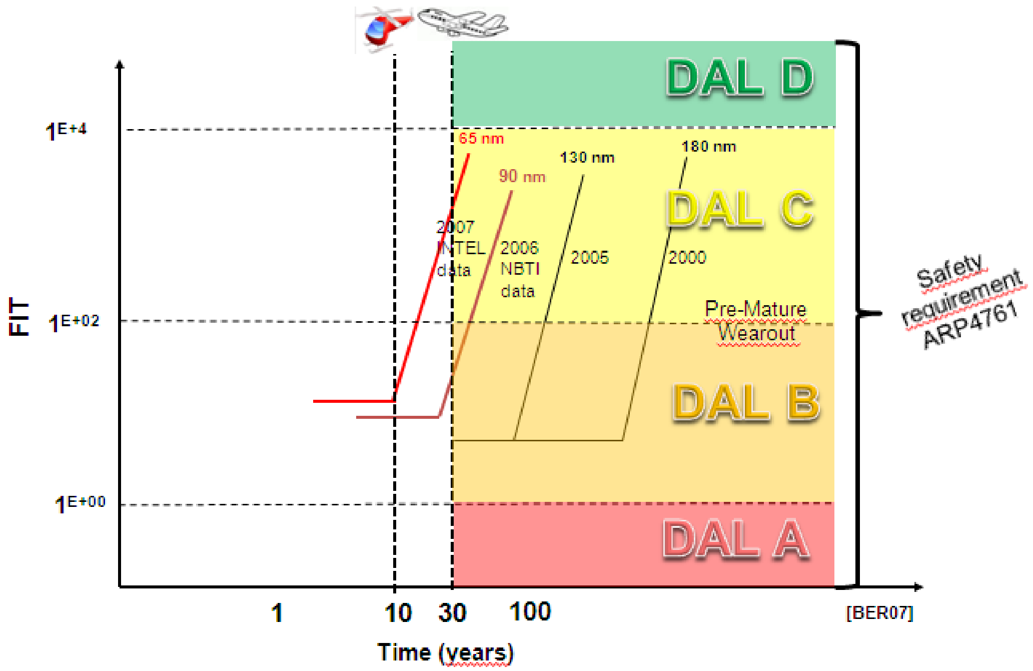

Figure 20. Evolution of the bathtub curve with technology [36]

5.4.2 Predicting Reliability Growth in Systems

This example is investigating the use of reliability models, tuned against historical test data, to support the Department of Defense requirement for reliability planning models to set intermediate MTBF objectives. The approach depicted below is that of L. Crow, which is one of several approaches currently in use [37, 38]. To be applicable, the system must be complex in the sense that the number of potential failures should be large enough to comply with the statistical structure described below.

The US Army Materiel Systems Analysis Activity (AMSAA) reliability growth model was developed in the 1970s using a statistical basis to support the application of the goodness-of-fit and confidence interval properties listed in this report. The statistical structure of the model is a non-homogeneous Poisson process (NHPP) model with a Weibull intensity function:

$$r(t) = \lambda \beta t^{\beta - 1} \tag{5}$$

where *r(t)* is the failure intensity, λ is a positive model parameter that affects scaling, and β is the other positive model parameter that shapes the intensity function (see figure 21):

- If β < 1, the failure intensity decreases, implying a reliability growth.
- If β > 1, the failure intensity increases, implying a decrease in system reliability.
- If β = 1, the model collapses to homogeneous Poisson or exponential distribution.

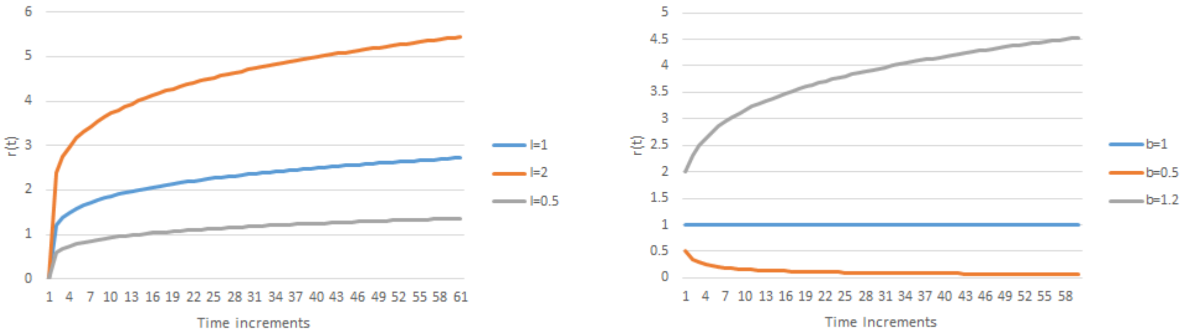

Figure 21. Effect of λ and β parameters on reliability growth model

The physical interpretation of β is the ratio of the cumulative MTBF to the instantaneous MTBF at time, *t*.

To provide more contextual information, this model is used upstream of the production phase, for which the system is deemed mature; that is, there is no further reliability improvement. The expected MTBF for the production system is therefore obtained from the predicted MTBF at the end of the testing phase. This prediction is achieved by using the AMSAA model tuned on grouped historical test data:

$$M\hat{T}BF = \frac{1}{\hat{r}(T)} = \frac{1}{\hat{\lambda}\hat{\beta}T^{\hat{\beta}-1}} \tag{6}$$

where *T* is the end of test time, $\hat{\lambda}$ the MLE of the scaling parameter, and λ and $\hat{\beta}$ the MLE of the shaping parameter, β.

The activities of reliability growth evaluation are similar to the ones discussed in this report, upstream of their use in programmatic considerations affecting the system production [38]. For example:

- The collection and analysis of test data (typically time-censored grouped data) to understand what is contained in the data and where the shortcomings are, identify the outliers that would affect the model parameter estimation, and identify the failure modes captured in the data
- The selection of distribution functions and functional models. Reliability growth models have typically followed the power law family of models, with refinements captured in the AMSAA described above.
- The development of system-level parameter estimates for the selected model based on the high-failure rate modes, severity (fault classification scheme), and cost. Earlier methods used the MSE method, whereas more recent ones, including AMSAA, use the MLE approach.
- The computation of confidence bounds for the estimates.
- The determination of compliance with the reliability objective requirement.

With a focus on step 2 and the selection of the model, consider the criteria for the intrinsic features to have allowed the narrowing down to the HPP and NHHP. The next step is to determine which of the family of models presents the better agreement with the data. This determination can be made both graphically and by using statistical tools. Consider the simple example data set created from data in reference 38: three systems have been subjected to testing wherein seven failures occurred at different rates within 410 test time units. The cumulative failure is plotted against the cumulative test time in figure 22.

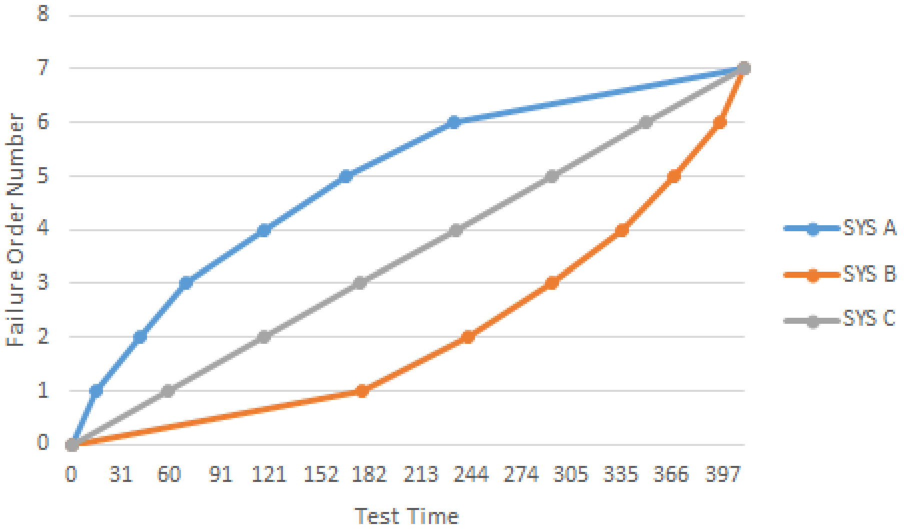

Figure 22. Cumulative failures against cumulative test time

The visual inspection is used to detect whether there is a trend in the occurrence of failures to select between HPP and NHPP and assess whether there is a system improvement or deterioration. One of the assumptions of HPP is that the times between failures are independent identically exponentially distributed, and NHPP allows for times between failures to increase or decrease with time. Both systems A and B show a variation with time of the occurrence of failures and, therefore, would be better modeled by NHPP, whereas system C exhibits regularly spaced failures in time and, therefore, would have a better fit with a HPP model. The trend in reliability growth is visualized through an increasing time interval between occurrences of failures, which shows as a convex shape. In opposition, system deterioration is visualized through the acceleration with time of failure occurrence and a concave shape of the failures versus time. System A will therefore be better modeled by NHPP with a positive growth rate (or $\beta < 1$), whereas system B will be modeled by NHPP with a negative growth rate (or $\beta > 1$). For more complex data sets, statistical tests exist to determine if times between failures increase, decrease, or remain constant (e.g., Laplace test).

5.4.3 Use of Reliability Model and Statistical Properties

This example aims at discussing the application of a model to collected data and what can be investigated from the statistical properties. The data used in this example are extracted from [38] and the determination of the model parameter estimates. The subsequent derivation of statistical properties and discussion is not from the handbook and is provided as example value.

Consider collected data of system failure (in the MTBF sense, the system no longer performs its intended function) over time. The data are shown in table 6 as a time-truncated sample after 300 hours of operation.

Table 6. Historical data of system failure [38]

Failure Order Number	Time of Occurrence [hour] (cumulative)	Failure Order Number	Time of Occurrence [hour] (cumulative)
1	2.6	15	98.1
2	16.5	16	101.1
3	16.5	17	132
4	17	18	142.2
5	21.4	19	147.7
6	29.1	20	149
7	33.3	21	167.2
8	56.5	22	190.7
9	63.1	23	193
10	70.6	24	198.7
11	73	25	251.9
12	77.7	26	282.5
13	93.9	27	286.1
14	95.5	End of sample	300

The first transformation is to formulate the data for reliability modeling (i.e., in the form of cumulative rate of occurrence of failures), as shown in figure 23.

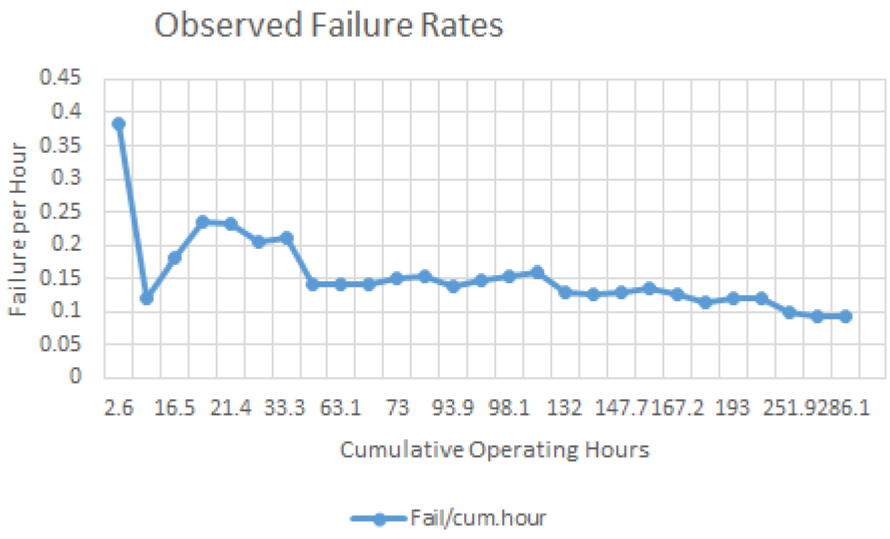

Figure 23. Observed system failure rate

Consider that the approach selected is with continuous time models, because the usage domain of the system at hand is continuous time (e.g., continuously operating during the data collection). Visual inspection of the graph in figure 23 directs the model selection toward NHPP (failures occur not regularly spaced in time; see previous example) with an exponential envelope. Remember that several models can fit the data, so the starting point may be the most used model of this kind—the Power Law

Process—which supports a parametric approach (meaning all statistical properties have a closed form; no numerical method is needed). The general form for the model is given by equation C-40 or, equivalently, equation C-41, both found in appendix C.

The next step is to determine the parameters in the model—namely, the scaling λ and the shape β—the maximum likelihood estimates yield:

$$\begin{aligned} \hat{\lambda} &= 0.404 \\ \hat{\beta} &= 0.826 \end{aligned} \tag{7}$$

The decreasing rate of failure is coherent with $\hat{\beta}$ being smaller than 1. Now the estimated failure rates can be plotted against the observed failure rates, as shown in figure 24.

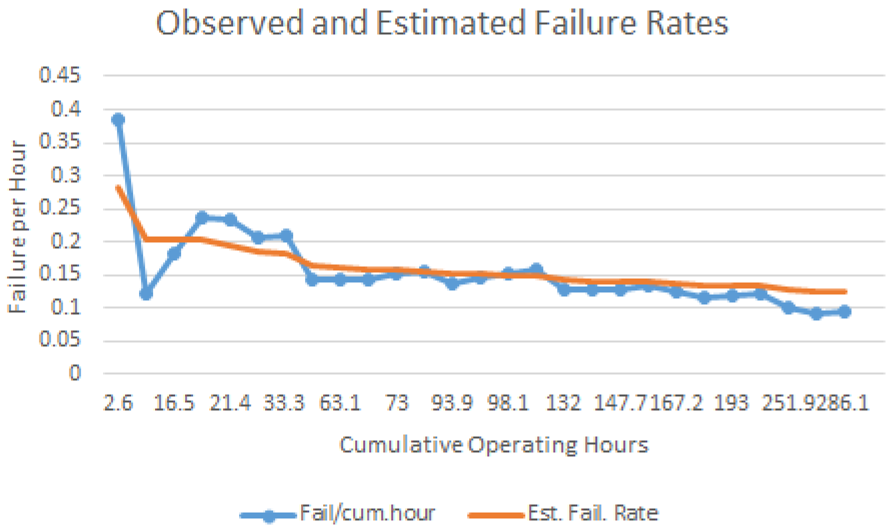

Figure 24. Observed and estimated system failure rates

The model fits most of the data as expected from a maximum likelihood approach. The sharp changes in the first 50 hours present a challenge for the model. Similarly, toward the end of the data, the model is weighted by the well-fitted data between 60 and 240 hours, so the start of a change in the slope of the failure rate is not well-captured.

The next step is to compute some of the statistical properties to provide a quantitative assessment of how well the model fits. This example will be limited to two easily understandable properties: 1) the estimator error (bias) and 2) estimator variance (standard deviation). The estimator error is the difference between the estimated failure rate and the observed failure rate. Visually, it is the gap between the red curve and the blue curve in figure 24. Ideally, the estimator variance should oscillate about a zero value, meaning that it does not specifically underestimate or overestimate the observations. By inspection, the red curve is more often above the blue curve than below, so a positive bias should be expected.

Figure 25 confirms the visual inspection with a positive bias of average value of 0.0057 failure per hour. The type of analysis that can be done on the bias is:

- On its magnitude–What is an acceptable error on the prediction of failure rate? The discussion can be based on the average bias (here, 0.6%) or using confidence bounds, such as the error is smaller than 5% for 98% of the time.

- On its trend–This is especially relevant because the model is likely to be used for prediction of reliability at some point in the future. The data show an increasing trend in the estimator error, but because the data are truncated, it is impossible to know whether the trend is temporary (e.g., could be outliers) or indicative of a change in shape in the failure rate function. Additional data (in the same conditions) would help rule out outliers or change in the model (e.g., adding a parameter) to specifically fit the last 50 hours and may be recommended.

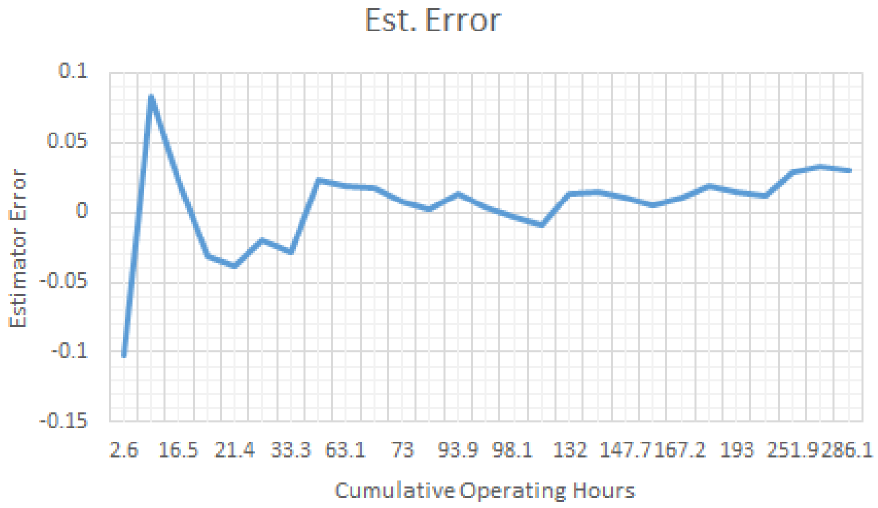

Figure 25. Evolution in time of estimator bias

The second property describes the dispersion of the estimates. A good estimator presents a small dispersion. Standard deviation is used as the measure of dispersion and is obtained from the differences between the estimated failure rates and average of the sample data of the estimated failure rate. Figure 26 indicates that the early estimates are “off,” but once the shape of the failure rate stabilizes past the 33rd operating hour, the estimates converge quickly and remain within 2% until the 150th operating hour.

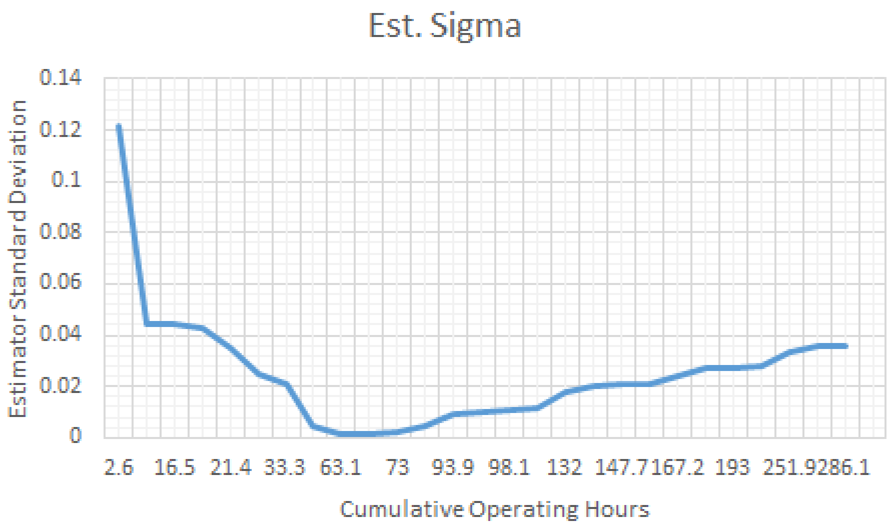

Figure 26. Evolution in time of estimator standard deviation

The analysis on the standard deviation is similar to that on the bias, namely:

- On the magnitude–σ-limits can be set even if the distribution is not normal.
- On the trend–The increase in the estimator error is captured earlier as a growing trend in the standard deviation. By construct, the standard deviation identifies differences between an estimate and the averaged value of the estimated sample data, whereas the bias is a point-to-point comparison.

6. FINDINGS FOR SYSTEM-LEVEL BLACK BOX APPROACH

Though the criteria and decision process discussed in sections 4 and 5 are valuable to understand the potential issues with service history or product experience data, they may be difficult to implement in a practical manner. This section proposes a complementary top-down method applicable at system level and where the system is considered to be a black box.

To provide a direct answer to the five questions in section 2, service history may be considered for use if an ELOS is provided by the applicant. The approach in this report proposed well-defined criteria based on the definition of EDAL and their ELS. The EDAL is associated with in-service data and, together with ELS, allows for claiming an ELOS. The approach uses the principles introduced by the systemic method, general system theory as defined in [39], and cybernetics as introduced in [40, 41]. The EDAL principle is defined in section 6.3 and based on the feedback loop described in section 6.2.

6.1 DEFINITIONS

6.1.1 Area of Use

An area of use is the domain in which the relationship between the system and the context in which the system will be involved continue to be understandable, predictable, and controllable.

6.1.2 Compliance

Compliance between the encompassing system need and area of use ensures that the emergent phenomena of the encompassing system remain understandable, predictable, controllable, and consistent with the safety in the context of the system under consideration.

6.1.3 Description

The definition of description is based on the concept defined by "definite description"—also called Russell's "theory of description"—which was defined by Bertrand Russell in his paper [42].

A description defines an exhaustive list of conditions to be adhered to by the users to ensure that the relationships between the described elements continue to be understandable, predictable, and controllable. Descriptions may define the meta-conditions (conditions to be adhered to by the usage of the list of conditions) associated with the list of conditions. Description may provide a variety of the described element.

6.1.4 Knowledge Domain

A knowledge domain defines an exhaustive list of descriptions to be taken into account by the user(s). Knowledge domain may define the meta-descriptions (descriptions to be taken into account by the usage of the list of conditions) associated with the list of conditions.

6.1.5 Variety

This concept was defined by Ross Ashby in his "An Introduction to Cybernetics" [43]. For contextual reference, this extract allows the user to clearly define the meaning of the term used:

"[…] a system has a variety of possible states of equilibrium, […]"

The word "variety," in relation to a set of distinguishable elements, will be used to mean […] the number of distinct elements.

6.2 FEEDBACK PRINCIPLE AND SAFETY ASSURANCE STANDARDS MODEL

Considering that the development process is a set of controlled activities, the feedback and error-controlled regulator principles, as introduced by Ashby and Wiener [40, 41], are used to model the safety assurance standards. Reference 44 provides relevant information to build the control abstraction model shown in figure 27.

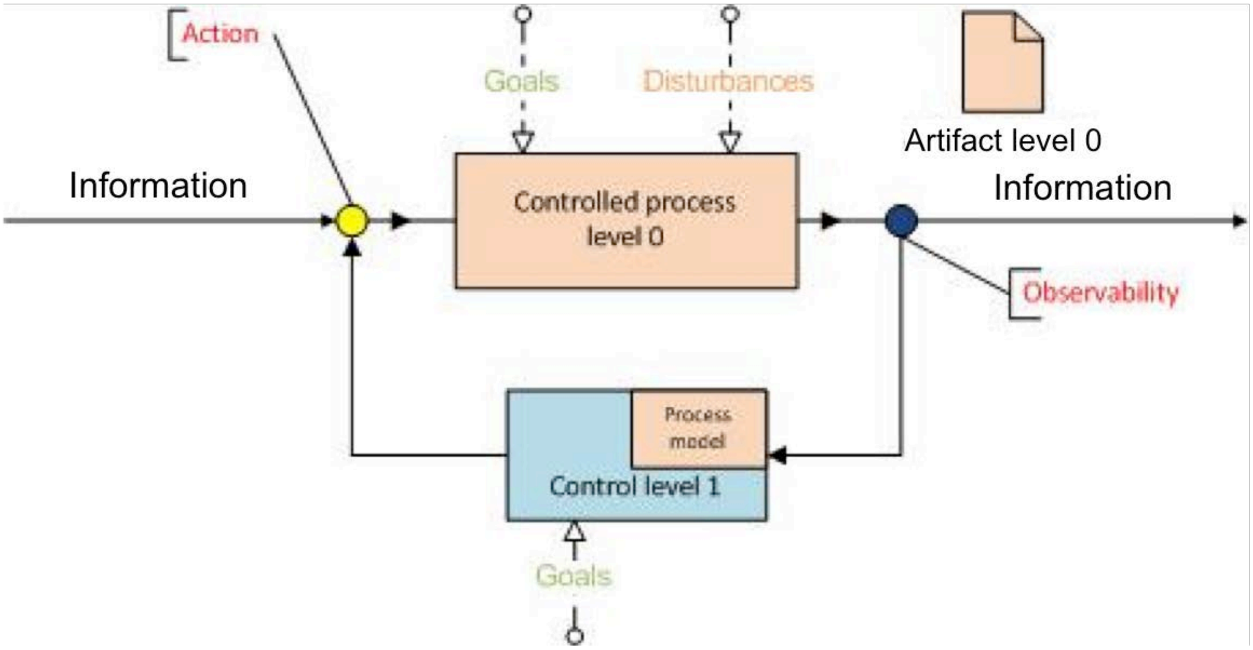

Figure 27. Feedback principle and safety assurance standard model

This model uses the following terms:

- Controlled process (level 0)–Development process used to build output information from inputs information with associated artifact defined by the development process itself to reach the goals and mitigate the disturbances. The intent of goals is to guide the decision-making process and break down goals into objectives to prescribe process steps that need to be taken to meet goals.
- Control (level 1)–Observation of the outputs information leads to the assessment of compliance of the development process, based on the associated controlled process description or process model, and prescribes corrective actions (positive, negative or null feedback) through inputs information for the controlled process.

A transformation model is needed to show the intents of the goals set by the safety assurance standards (“transformation,” as defined in [45], and “model,” as defined in [46], for which the abstract representation is the transformation). As defined in [47], safety assurance standards are prescriptive on the processes and model representation is a common way to explain the intended use of safety standards.

6.2.1 Concept of Abstraction

The concept of abstraction is used as a technique for managing the structure and organizational levels of the considered systems. The principle is to establish a level of organization at which a person interacts with the system and to suppress the structural or organizational details below the level of organization at which a person interacts with the system called the current level.

The abstraction concept deals with the concept of description, as defined in section 2.1.1.2, and should be considered in the case of the law of requisite variety, as defined in [48], to manage complexity. The relationship between the description concept and complexity management will be addressed later with the concept of complex system.

6.2.2 Control Abstraction and Information Abstraction

Abstraction can apply to control or data: control abstraction is the abstraction of actions structures, whereas data abstraction is that of data structures:

- Control abstraction involves the use of subprograms and related control flows concepts.
- Data abstraction allows for handling data bits in meaningful ways and related data flows concepts.

Based on general system theory, described in [39], information abstraction is used instead of data abstraction. The advantage to this is in the improving and underlining of the relationship between the feedback principle previously explained and the general system theory method and cybernetics using information concept rather than data concept.

Two kinds of abstraction are used to represent control flow and information flow:

- Model based on control abstraction (MBCA)–The purpose of this kind of model is to highlight control flow and constituent parts of the control flow that are associated with the concept of information transformation and combination. In this model, information is represented by the links between control abstractions (information is an input or an output of the control abstraction), as shown in figure 28.

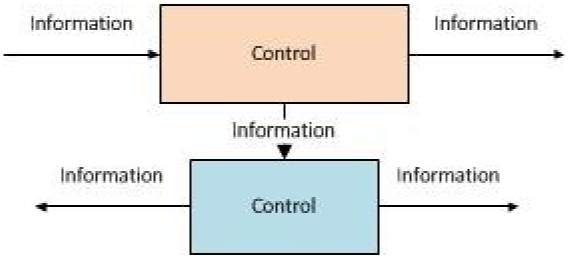

Figure 28. Control abstraction representation

- Model based on information abstraction (MBIA)–The purpose of this kind of model is to highlight information flow and constituent parts of the information flow that are associated with the concept of system state transformation and combination. In this model, a control flow is represented by the links between information abstractions (control is an input or an output of the information abstraction) [45], as shown in figure 29.

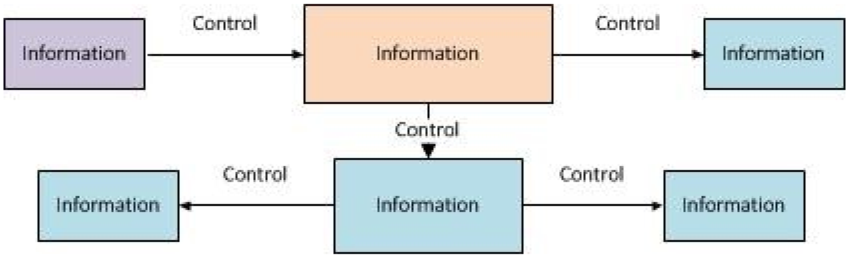

Figure 29. Information abstraction layer

6.2.3 Intended Function of the Safety Assurance Standards

Section 6.2.2 offers the basic definitions necessary to identify the intended function of safety assurance standards.

The objectives prescribed by safety assurance standards can be classified using a method close to that suggested by TIAM in [22]. Criteria related to controlled process and control goals need to be expressed according to the definitions in section 6.1. These criteria should help clearly identify goals and objectives associated with the intended function definition of the processes of those associated with quality scope of the control. The keywords associated with this criterion are compliance, ensure, and conformance. In the scope of this approach, other objectives address other kinds of goals (e.g., life-cycle data properties or dedicated technique of the process or method) and, therefore, are not used to identify the intended function of the safety-assurance standards.

Objectives identified as relevant to be linked with controlled process and control goals are listed in the following sections and organized by safety standards and associated abstraction levels.

6.2.3.1 DO-178C Controlled Process, Control Goals, and Software Development System State

Applying criteria identified in section 2.1.5, a study of the DO-178C software guidance may provide an MBIA perspective of the software development process and the software verification process, as shown in figure 30.

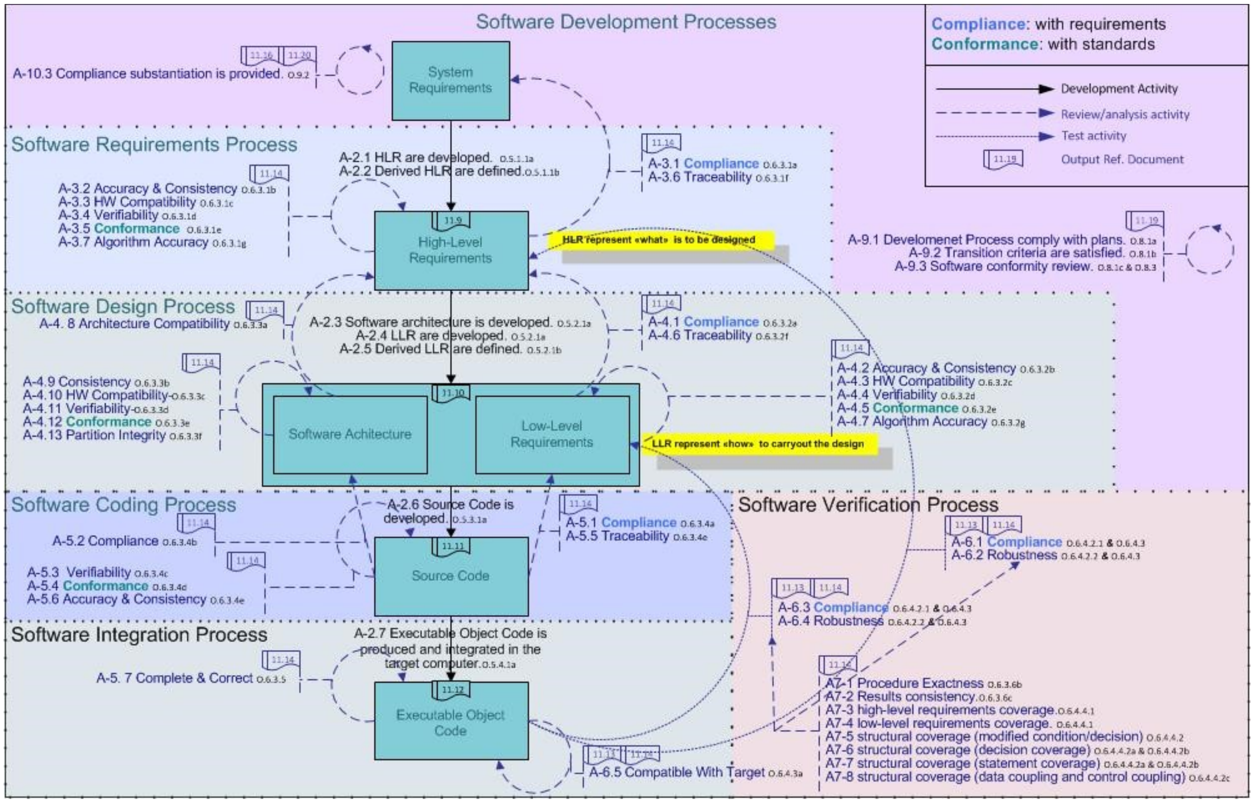
Software Development Processes
Compliance: with requirements
Conformance: with standards
Development Activity
Review/analysis activity
Test activity
11.19
Output Ref. Document
11.16
11.20
A-10.3 Compliance substantiation is provided. O.9.2
System Requirements
Software Requirements Process
A-2.1 HLR are developed. O.5.1.1a
A-2.2 Derived HLR are defined. O.5.1.1b
11.14
A-3.1 Compliance O.6.3.1a
A-3.6 Traceability O.6.3.1f
11.14
A-3.2 Accuracy & Consistency O.6.3.1b
A-3.3 HW Compatibility O.6.3.1c
A-3.4 Verifiability O.6.3.1d
A-3.5 Conformance O.6.3.1e
A-3.7 Algorithm Accuracy O.6.3.1g
11.9
High-Level Requirements
HLR represent «what» is to be designed
11.19
A-9.1 Develomenet Process comply with plans. O.8.1a
A-9.2 Transition criteria are satisfied. O.8.1b
A-9.3 Software conformity review. O.8.1c & O.8.3
Software Design Process
11.14
A-4. 8 Architecture Compatibility O.6.3.3a
A-2.3 Software architecture is developed. O.5.2.1a
A-2.4 LLR are developed. O.5.2.1a
A-2.5 Derived LLR are defined. O.5.2.1b
11.14
A-4.1 Compliance O.6.3.2a
A-4.6 Traceability O.6.3.2f
11.14
A-4.9 Consistency O.6.3.3b
A-4.10 HW Compatibility-O.6.3.3c
A-4.11 Verifiability-O.6.3.3d
A-4.12 Conformance O.6.3.3e
A-4.13 Partition Integrity O.6.3.3f
11.10
Software Achitecture
Low-Level Requirements
11.14
A-4.2 Accuracy & Consistency O.6.3.2b
A-4.3 HW Compatibility O.6.3.2c
A-4.4 Verifiability O.6.3.2d
A-4.5 Conformance O.6.3.2e
A-4.7 Algorithm Accuracy O.6.3.2g
LLR represent «how» to carryout the design
Software Coding Process
A-2.6 Source Code is developed. O.5.3.1a
11.14
A-5.1 Compliance O.6.3.4a
A-5.5 Traceability O.6.3.4e
11.14
A-5.2 Compliance O.6.3.4b
11.14
A-5.3 Verifiability O.6.3.4c
A-5.4 Conformance O.6.3.4d
A-5.6 Accuracy & Consistency O.6.3.4e
11.11
Source Code
Software Verification Process
11.13
11.14
A-6.1 Compliance O.6.4.2.1 & O.6.4.3
A-6.2 Robustness O.6.4.2.2 & O.6.4.3
11.13
11.14
A-6.3 Compliance O.6.4.2.1 & O.6.4.3
A-6.4 Robustness O.6.4.2.2 & O.6.4.3
Software Integration Process
A-2.7 Executable Object Code is produced and integrated in the target computer. O.5.4.1a
11.14
A-5. 7 Complete & Correct O.6.3.5
11.12
Executable Object Code
11.14
A7-1 Procedure Exactness O.6.3.6b
A7-2 Results consistency. O.6.3.6c
A7-3 high-level requirements coverage. O.6.4.4.1
A7-4 low-level requirements coverage. O.6.4.4.1
A7-5 structural coverage (modified condition/decision) O.6.4.4.2
A7-6 structural coverage (decision coverage) O.6.4.4.2a & O.6.4.4.2b
A7-7 structural coverage (statement coverage) O.6.4.4.2a & O.6.4.4.2b
A7-8 structural coverage (data coupling and control coupling) O.6.4.4.2c
11.13
11.14
A-6.5 Compatible With Target O.6.4.3a

Figure 30. MBIA of DO-178C software development process

Using this MBIA representation, it is possible to identify the objectives linked only to life-cycle data related to the MBIA perspective and therefore related to the software development system state. The result of this identification is presented in table 7.

Table 7. Software development system state

Goal Description	Software Development System State	System State Observability
Software plans comply with DO-178C	Dev. Plan	ABCD
HLRs conform to standards	HLR	ABC
HLRs are compatible with target computer	HLR	AB
HLRs comply with system requirements	HLR	ABCD
LLRs conform to standards	LLR	ABC
LLRs are compatible with target computer	LLR	AB
LLRs comply with HLRs	LLR	ABC
Software architecture conforms to standards	Architecture	ABC
Software architecture is compatible with target computer	Architecture	AB
Software architecture is compatible with HLRs	Architecture	ABC
Source code conforms to standards	Code	ABC
Source code complies with software architecture	Code	ABC
Source code complies with LLRs	Code	ABC
EOC complies with LLRs	EOC	ABC
EOC is compatible with target computer	EOC	ABCD
EOC complies with HLRs	EOC	ABCD
Assurance is obtained that software development and integral processes comply with approved software plans and standards	Assurance of Dev. Plan	ABCD

HLR = high-level requirement; LLR = low-level requirement

Thanks to this abstraction representation, regarding the safety assurance standards and feedback principle provided in section 2.1.2, a strong analogy is established between, on the one side, the development process and controlled process, and, on the other, the verification process and control process.

In table 7 and based on DO-178C content, software development system states identified from the associated MBIA are high-level requirement (HLR), low-level requirement (LLR), development plan (Dev. Plan), architecture, code, executable object code (EOC), and assurance development plan. Input state is system requirements, and the end state is EOC. System state observability is defined by the assurance level for which the system state is required; for example, if assurance level A, B, C, and D require the system state, system state observability is ABCD.

6.2.3.2 ARP-4754A and DO-297 Controlled Process and Control Goals

Applying the criteria identified in section 6.1, safety assurance standards for system prescribes the objectives linked only to life-cycle data related to the MBIA perspective and, therefore, related to the system development system state (see table 8).

Table 8. System development system state

Goal Description	System Development System State	System State Observability
System plans comply with AMC	Dev. Plan	ABCD
System-level specifications conform to specification common language (information standard)	Syst. Level specifications	ABC
System-level specifications are compatible with target items	Syst. Level specifications	AB
System-level specifications comply with aircraft specifications	Syst. Level specifications	ABCD
System architecture conforms to standards	Syst. Architecture	ABC
System architecture is compatible with target items	Syst. Architecture	AB
System architecture is compatible with system-level specifications	Syst. Architecture	ABC
Items comply with system-level specifications	Syst. Item	ABCD
Assurance is obtained that system development and integral processes comply with approved plans and standards	Assurance of Dev. Plan	ABCD

For each item (hardware, software, or other), system specifications include the environmental requirements upon which the system is designed.

6.3 SYSTEM MODEL, PERCEPTION, AND EDAL

6.3.1 System Model and Perception Level

A system MBIA allows a representation of the perception of the system or part of the system to be constructed. Depending on the level, the perception representation should be detailed, as described in the sections that follow.

6.3.1.1 Perception Level 1

Figure 31 shows the system states for Perception Level 1 (PL1).

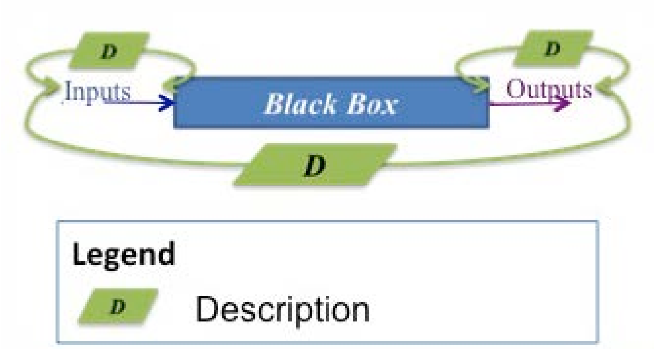

Figure 31. PL 1

PL1 should provide:

- A knowledge domain development plan, including service history information and data use.
- A black box description of the relationship between inputs and outputs.
- A description of inputs and relationships between the constituent parts of the inputs.
- A description of outputs and relationships between the constituent parts of the outputs.
- Assurance that knowledge domain development and integral processes are compliant with approved plans and standards.

6.3.1.2 Perception Level 2

Figure 32 shows the system states for Perception Level 2 (PL2).

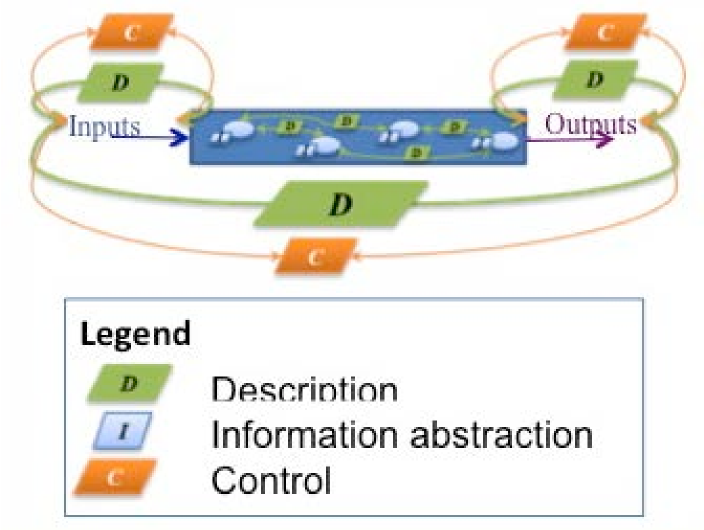

Figure 32. PL 2

PL2 should augment PL1 with the following:

- Conformity with the standard used for expressing the description.
- Description of perceptible constituent parts of the black box.
- Description of the functional connections of the relational imbrications of the constituent parts of the black box.

6.3.1.3 Perception Level 3

Figure 33 shows the system states for Perception Level 3 (PL3):

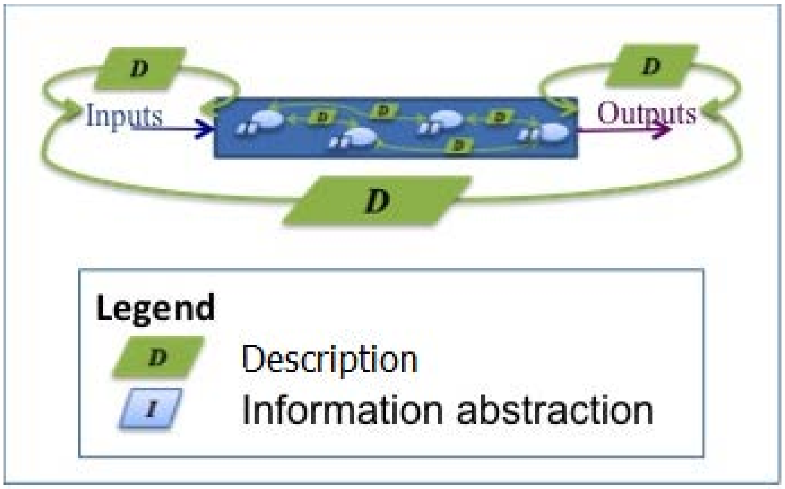

Figure 33. PL 3

PL3 should augment PL2 with the following:

- Control loop of the relationship between the constituent parts of the inputs.
- Control loop of the relationship between the constituent parts of the outputs.
- Control loop of the relationship between the inputs and outputs.

6.3.2 System Model and Equivalent Level of Design Assurance

The model of a system and associated PL provided depend directly on the observer who perceives the system; the observer has a relationship with the system called influence on the perception of a system. To mitigate the influence of the observer in the perception of the system and, in particular, to mitigate the Whorfian hypothesis defined in [39], the system descriptions should be developed from a shared vocabulary between stakeholders. In case of MBCA, the dictionary should address the control definition and usage; in case of MBIA, the dictionary should address the information definition and usage.

Analogous to safety standards, the system modeling development needs independence. This independence is allocated to system description independence, and system perception independence is associated with PL and EDAL, as defined in table 9.

Table 9. EDAL and system PL

EDAL	PL	PL Developed with Independence
A	PL3	PL2
B	PL3	PL1
C	PL2	PL1
D	PL1	N/A

The accuracy in the perception of the system (development of a mental model, as defined in [49]) must increase in consistency with the safety objective of the encompassing system in which the system is, and will be, involved. Developed as a whole, a system perception model should distinguish two groups of characteristic features: a structural group of characteristic features and functional group of characteristic features, as suggested in [50]. These two groups of characteristics should be linked to behavior and resources they mobilize for their interactions.

6.3.2.1 Definition of EDAL Concept Using Service History

Service history/product experience data allow for the assertion/claim of an EDAL according to the following product characteristics:

- EDAL-A–The product is identifiable and only performs the intended function without any fault or failure regarding the expected needed performances in the intended operational environment and according to PL3 and an independent PL2, as defined above.
- EDAL-B–The product is identifiable and only performs the intended function with known minor faults or failures regarding expected needed performances in the intended operational environment, not exceeding a predefined amount of faults or failures, and according to PL3 and an independent PL1, as defined above.
- EDAL-C–The product is identifiable and performs the intended function among others, with known minor faults or failures regarding expected needed performances in the intended operational environment, not exceeding a predefined amount of faults or failures, and according to the PL2 and an independent PL1, as defined above.
- EDAL-D–The product is identifiable and performs the intended function among others and according to the PL1, as defined above.
- EDAL-E–The product is identifiable.

6.3.2.2 Application to Software Development System

The aforementioned EDAL definitions provide an equivalent goal traceability with respect to the software development system state compliant with DO-178C. Table 10 is a proposal by the authors of this report.

Table 10. Equivalent goal traceability with respect to system state and PL

Goal Description	Software Development System State	System State Observability	Minimum PL Coverage
Software plans comply with DO-178C	Dev. Plan	ABCD	PL1
HLRs conform to standards	HLR	ABC	PL2
HLRs are compatible with target computer	HLR	AB	PL3
HLRs comply with system requirements	HLR	ABCD	PL1
LLRs conform to standards	LLR	ABC	PL2
LLRs are compatible with target computer	LLR	AB	PL3
LLRs comply with HLRs	LLR	ABC	PL2
Software architecture conforms to standards	Architecture	ABC	PL2
Software architecture is compatible with target computer	Architecture	AB	PL3
Software architecture is compatible with HLRs	Architecture	ABC	PL2
Source code conforms to standards	Code	ABC	PL2
Source code complies with software architecture.	Code	ABC	PL2
Source code complies with LLRs	Code	ABC	PL2
EOC complies with LLRs	EOC	ABC	PL2
EOC is compatible with target computer	EOC	ABCD	PL1
EOC complies with HLRs	EOC	ABCD	PL1
Assurance is obtained that software development and integral processes comply with approved software plans and standards	Assurance of Dev. Plan	ABCD	PL1

6.4 DETERMINING ELS AND ELOS

The approach described in the previous sections provides well-defined criteria based on EDAL. This section defines their ELS, allowing for claiming an ELOS.

6.4.1 The ELS

Service experience information should be analyzed to determine the failure condition severity classification for which the reused part of the product is used and its contribution in the implementation of the related intended function or functionality. This analysis provides the ELS of the reused part of the product against the following classifications: ELS-Catastrophic, ELS-Hazardous/Severe Major, ELS-Major, ELS-Minor, or ELS-No Safety Effect.

6.4.2 The ELOS

The EDAL level in conjunction with the ELS level allow for the assertion/claim of an associated ELOS according to FAA Order 8110.112 [51] and to reuse the part of the product with this ELOS within the context specified in the ELS memorandum. The recommended credit typology and credit level for the reuse are shown in table 11.

Table 11. Recommended credit topology using EDAL, ELS, and ELOS

EDAL	ELS	ELOS
EDAL-A	ELS-Catastrophic	Catastrophic
EDAL-B	ELS-Hazardous/Severe Major	Hazardous/Severe Major
EDAL-C	ELS-Major	Major
EDAL-D	ELS-Minor	Minor
EDAL-E	ELS-No Safety Effect	No Safety Effect

6.5 ASSESSMENT OF SUFFICIENCY OF SERVICE HISTORY/PRODUCT EXPERIENCE

The relationship between severity of failure condition and allowable quantitative probability by aircraft type is driven by associated certification specifications. Considering table 12 as an example, to classify failure condition to its occurrence probability per flight hour (applicable only to transport-category aircraft), it is possible to provide a sufficient assessment of service history or product history period per flight hour. The probabilities of failure per flight hour in table 12 were derived for random failures. In the proposed approach, it is assumed—in the absence of relevant statistics—that design faults occur as often as random faults. Therefore, table 12 probabilities can be reused.

Table 12. Failure condition classification for transport-category aircraft

	Aircraft type	Failure Condition Classification			
FAA	Transport category AC 25.1309-1A	Minor	Major-significant	Severe Major	Catastrophic
FAA	Transport category AC 25.1309-1A Probability per flight hour	Probable $P > 10^{-5}$	Improbable $10^{-9} < P < 10^{-5}$		Extremely improbable $P < 10^{-9}$
FAA	AC 20-174 (ARP4754A)	Development assurance process assigned level: D	Development assurance process assigned level: C	Development assurance process assigned level: B	Development assurance process assigned level: A
EASA	Large aeroplane AMC 25.1309 Failure conditions	Minor	Major	Hazardous	Catastrophic
EASA	Large eeroplane–AMC 25.1309-Probability per flight hour	Probable $10^{-5} < P < 10^{-3}$	Remote $10^{-7} < P < 10^{-5}$	Extremely Remote $10^{-9} < P < 10^{-7}$	Extremely Improbable $P < 10^{-9}$
EASA	Development assurance design (AMC 25.1309 3b(2); ARP4754A)	Development assurance process assigned level: D	Development assurance process assigned level: C	Development assurance process assigned level: B	Development assurance process assigned level: A

If a manufacturer can substantiate the following:

1) The contribution of the particular system for which service history is claimed to a specific failure condition.
2) The contribution of the reused component within the system to the service history claim.

Then probability of failure for the reused part of the product P_R must meet:

$$P_R < \frac{P_s(FC)}{C_R} \tag{8}$$

where $P_S(FC)$ is the probability of failure at system level associated with the failure condition FC (see table 12), and C_R is the contribution of the reused part of the product to the failure condition.

The derivation and substantiation of C_R may be difficult and heavily relies on the data collection process. Though unlikely, if an applicant is able to provide a value for C_R, the service history credit claim could be considered but with no guarantee of acceptance. The minimum service history needed to demonstrate the probability of failure can then be formulated as:

P_R can be substantiated by demonstrating no more than one failure per $\dfrac{C_R}{P_S(FC)}$

Moreover, the EDAL is an equivalent credit from service history. To represent this discrepancy with scale, shifted graduations would be introduced between F-I DAL and EDAL, as shown in figure 34.

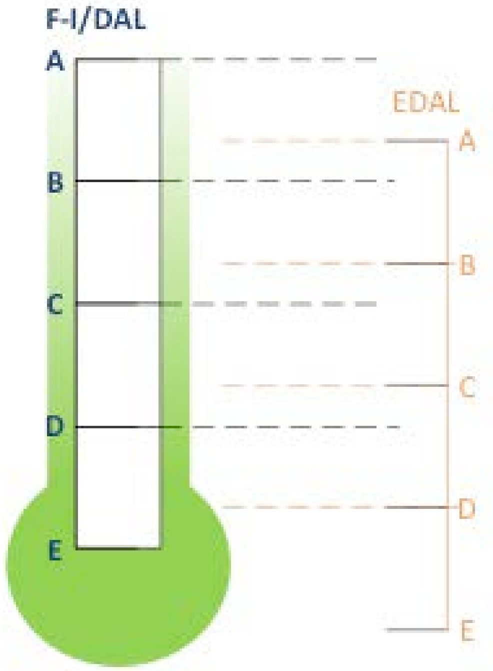

Figure 34. Proposed graduation shift between F-I DAL and EDAL

To claim compliance with F-I DAL and the associated probability of design fault from EDAL, verification of the associated PL is requested by using the appropriate AMC, as shown in figure 35.

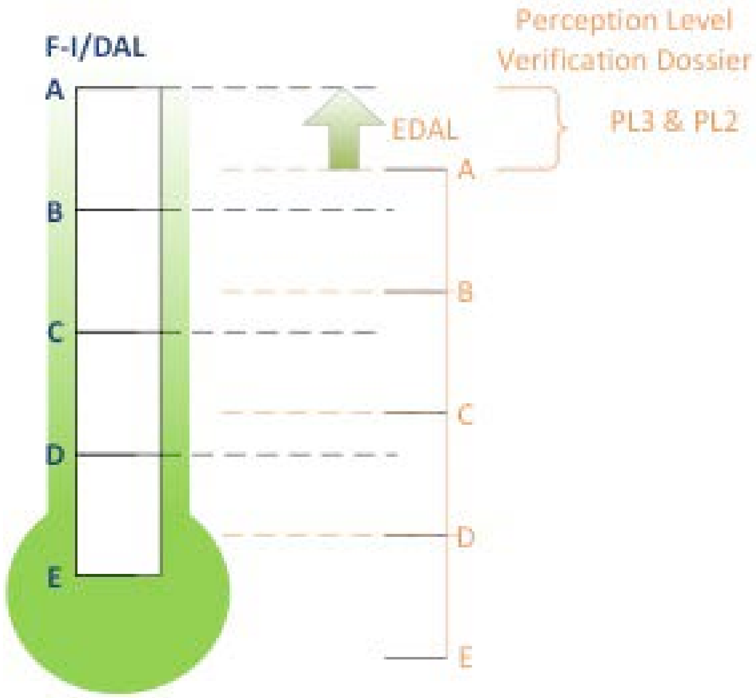

Figure 35. Articulation of EDAL and PL verification to meet F-I DAL

Table 13 presents a similar view in a tabular format.

Table 13. Design assurance credit and service history period

F-I Design Assurance Credit	Assessment of Service History Period in Equivalent Flight Hour	EDAL	ELS	ELOS
F-I/DAL-A	At least C_R/P_S(CAT)	EDAL-A	ELS-Catastrophic	Catastrophic
F-I/DAL-B	Between at least C_R/P_S(HAZ) and C_R/P_S(CAT)	EDAL-B	ELS-Hazardous/ Severe Major	Hazardous/ Severe Major
F-I/DAL-C	Between at least C_R/P_S(MAJ) and C_R/P_S(HAZ)	EDAL-C	ELS-Major	Major
F-I/DAL-D	Between at least C_R/P_S(MIN) and C_R/P_S(MAJ)	EDAL-D	ELS-Minor	Minor
F-I/DAL-E	No service history period	EDAL-E	ELS-No Safety Effect	No Safety Effect

CAT = category

6.6 CONSIDERATIONS REGARDING PROBLEM REPORTING

Regarding problem reporting during the service history period used to claim design assurance credit, and as described in section 6.5, the following classification should be addressed in the problem report system (though not yet actually performed in the current process):

- Problem report category (CAT)-I for issues having a compliance impact on the current means of compliance–A structured classification has to be proposed in this CAT based on the current existing means of compliance.
- Problem report CAT-II for issues having no impact on compliance of the current means of compliance–A classification should also be proposed in this CAT.

CAT-I issues should be considered for service history eligibility regarding the EDAL of the approach, particularly in view of the PL used. CAT-II issues may be considered to upgrade and improve the efficiency of the current means of compliance.

6.7 INFORMATION TO BE PROVIDED IN SUPPORTING DOCUMENTATION

Information provided for certification should include:

- The part of the product being reused in a manner that is coherent with the PL used to claim the EDAL.
- The domain in which in-service data were captured (e.g., public domain, railways, automobile, or military).
- A minimal set of relevant data should be identified through a predefined process. Those data should be the minimum allowed skeleton for service history eligibility.
- A description of the context of data collection, including the quality process used to capture these data.
- A description of the process used to monitor the continuing performance of the item in service (e.g., system, LRU, or component).
- A description of the problem correction process throughout the data collection period in a coherent manner with the EDAL being claimed.
- The characteristics of recorded information, including the data capture frequency and related issue reports.

6.7.1 Estimated Product Part Typology and Complexity

The diversity of solutions in aeronautics, and particularly the avionics field, led to the emergence of the notion of complexity (see glossary, section 8). The intent of this approach is to provide a PL of the system under consideration, regardless of the means used to build that system. These means can indeed introduce the distinction of complex systems, though the concept presented in this section does not have to be connected to the notion of a complex system, which is a choice made by the system designer.

Out of the possible definitions of complexity, the one that best matches the approach defines a complex system as a system whose overall PL is characterized by a reduced predictability. The recommendation is to reduce system complexity to improve predictability of the PL. The law of requisite variety used in this approach provides a means to fix the upper limit related to the use of complex systems in service history: Only the PL providing predictability in use of the system behavior is acceptable [48].

6.7.2 Complexity and Software-Based Adaptive Systems

Adaptive systems have the ability to adapt their behavior, in response to evolutions in the operational environment, to reach their essential goals. Currently, the solutions used to develop a non-simple system are based on the structuring and flowing dual system functions. The first of the dual system functions is the structuring information function that provides the governance rules and is included in the system evolution memory. The second function is the flowing information function that provides the performance rules for exchanges between, and within, systems. Recommendations for the flowing information function of a software-based adaptive solution are provided in the report on verification of adaptive systems [52].

6.7.3 Concept of Simplexity

Simplexity is defined as the means to provide complementary relationships between simplicity and complexity [53]. Simplexity is identified for complex system design and management as a means to reduce complexity. In biology, the concept of simplexity applies to the set of solutions found by living organisms to deal with information and situations while taking into account past experiences and anticipating future ones. Such solutions are new ways of addressing problems so that actions may be taken more quickly, more elegantly, and more efficiently. In relation to the proposed approach, the recommendation is to use the concept of simplexity to reduce the complexity of the PL.

6.7.4 Complexity and Statistical Approach

This proposed approach should be used to provide an associated Markov chain to the system or reused part of the product. As mentioned in EASA AMC 25.1309 [54], it is possible to use a Markov analysis to reach the safety objectives associated with a catastrophic failure condition. The recommendation is to use the Markov analysis with service history information to improve the confidence on the PL of the system or reused part of the product.

7. RECOMMENDATIONS, CONCLUSIONS, AND OPEN DISCUSSION ITEMS

7.1 SUMMARY OF FINDINGS

7.1.1 Data Collection

For the data collection to be effective with respect to using software service history or AEH service experience to claim certification credit, it must ensure that relevant elements are collected from the end user (airline) to the avionics manufacturer. The bottom-up approach requires more information in both volume and detail to cover the qualitative and quantitative criteria in sections 4 and 5. The top-down approach requires higher-level information to substantiate the statistics of minimum equivalent flight hours (EFH) per functional or item DAL.

These aspects should be integrated in the design phase of the product (software or hardware), because service history, or AEH service experience as an afterthought, is unlikely to meet the data needs. Improvements include continued deployment of structured failure reporting (e.g., FRACAS), further integration of software service history/AEH service experience considerations in the onboard maintenance system design requirements, standardization of fault classification, and guidelines to maintenance operators would help address the current issues with the data collection process (see section 3).

7.1.2 Approaches to Determining Suitability

This report presents two approaches:

- A bottom-up approach using qualitative and quantitative criteria that further the guidelines of software service history in RTCA DO-178C and can be matched to the guidelines for AEH service experience in RTCA DO-254 (appendix A).
- A top-down approach matching software service history or AEH service experience to EDAL, which, combined with a system-level ELS, allows for claiming an ELOS.

The bottom-up approach requires detailed information for its qualitative criteria and uses reliability modeling for its quantitative criteria. It can be formulated as a decision flow for the applicant, in which most of the decision branches lead to discarding software service history or AEH service experience as an alternative means to meet safety objectives.

The top-down approach is straightforward in proposing minimum equivalent flight hours as a function of DAL. The underlying quantitative criteria are being integrated in the determination of EDAL and ELS, whereas the qualitative criteria and information requirements are integrated in the connection between EDAL and system description.

7.1.3 Applicability or Usability

The bottom-up approach has limitations on applicability because it requires a lower level of details that would more likely lead to discarding the use of the software service history or AEH service experience. However, because it matches the existing guidelines, it is the approach currently being used by applicants in one form or another.

The top-down approach is applicable to all DAL and can be transferred to parts other than part 25 if the equivalent set of failure statistics can be defined. This approach is not yet implemented by industry but promises to open the current limitations on usability of software service history or AEH service experience for the applicant, while allowing the bottom-up approach to coexist. Its straightforward criteria would also clarify acceptability of the certification credit claim by the authority.

7.1.4 Researchers’ Viewpoint

As this report shows, software service history and AEH service experience have not reached the same level of usability in certification projects. Currently, only AEH service experience for DAL D products is being considered, whereas software service history is not. Therefore, no actual data of software service history or software reliability could be used in the report. Most examples are AEH-related or at system level where software and hardware items are not distinguished. The culprit lies with reliability modeling and the lack of industry consensus regarding what constitutes an acceptable quality of fit metric so that the reliability model can be used for predictive MTBF. Researchers with expertise in software or AEH often have different points of view on applicability.

The proposed approach using EDAL and ELS is promising and would benefit from further research so that the system description tied to EDAL can be implemented in the industry processes.

7.2 RECOMMENDATIONS FOR BOTTOM-UP APPROACH

The recommendations extracted from the findings are organized in line with the approach described in section 2. The recommendations take the form of question-based decision tree diagrams that support the applicant/designee in the assessment of the data collected in service around the following three themes:

- The suitability of the data.
- The similarity of the operating environment.
- The quantification of reliability.

7.2.1 Determining Suitability of Product Service Experience Data

The determination of suitability for product service experience data is based on qualitative and quantitative criteria defined in sections 4 and 5. The analysis of each of these criteria in context of relevance and sufficiency may orient the strategy for claiming certification credit to proceed according to the following steps:

- Use criticality and user impact qualitative criteria.

These criteria allow the direct investigation of the available in-service environment if the product is non-safety critical, or if any failure of this product is not visible or has a minor impact on the users. On the contrary, safety-critical and major user impacts may require the user to further investigate the product using the other criteria: level of innovation, complexity, and process considerations (i.e., available life-cycle data and activities on the product).

- Use quantitative environment similarity criterion.

Depending on the result of the qualitative criteria analysis, the in-service operational environment may be used “as is” or mixed with an acceptable operational environment; this mixed environment may involve several possibilities for the product configuration, thereby increasing the relevance of the service history quantitative data. Further recommendations to proceed with the assessment of similarity in operating environment are collected in section 7.2.2.

- Evaluate available quantitative data.

After having defined a suitable operational environment, the type and amount of quantitative data have to be established for the product or subparts of the product. If these defined quantitative data are available within the existing service history, they may be relevant if the product is correctly managed under configuration; events and problem reports are efficiently fed back and analyzed; and the feedback process between user and manufacturer is correctly applied.

Otherwise, there may be a need to create a “new service history” from an in-house set of simulators and test benches; in those cases, the terminology “service history” may be replaced by, for example, specific endurance testing or robustness testing, which will help in addressing the lack of effective in-service history while still being used as an alternative method for certification credit.

- Collecting in-service data.

Defining the data collection requirements is very important to ensure that sufficient and suitable data are collected by the users. Methodology and requirements for in-service data collection were developed as part of the FIDES research program [30]; the deliverables document which information should be required as part of the in-service reports to support the usability of service experience data in certification projects. Though it is ultimately up to the manufacturer to develop these reports, the recommended information to be provided is summarized in appendix D.

7.2.1.1 Proposed Decision Flow Diagram

The above steps can be aggregated in the form of a decision flow diagram, as shown in figure 36. Note that, from the researchers' standpoint (see section 7.1.4), this approach is currently limited to DAL D products.

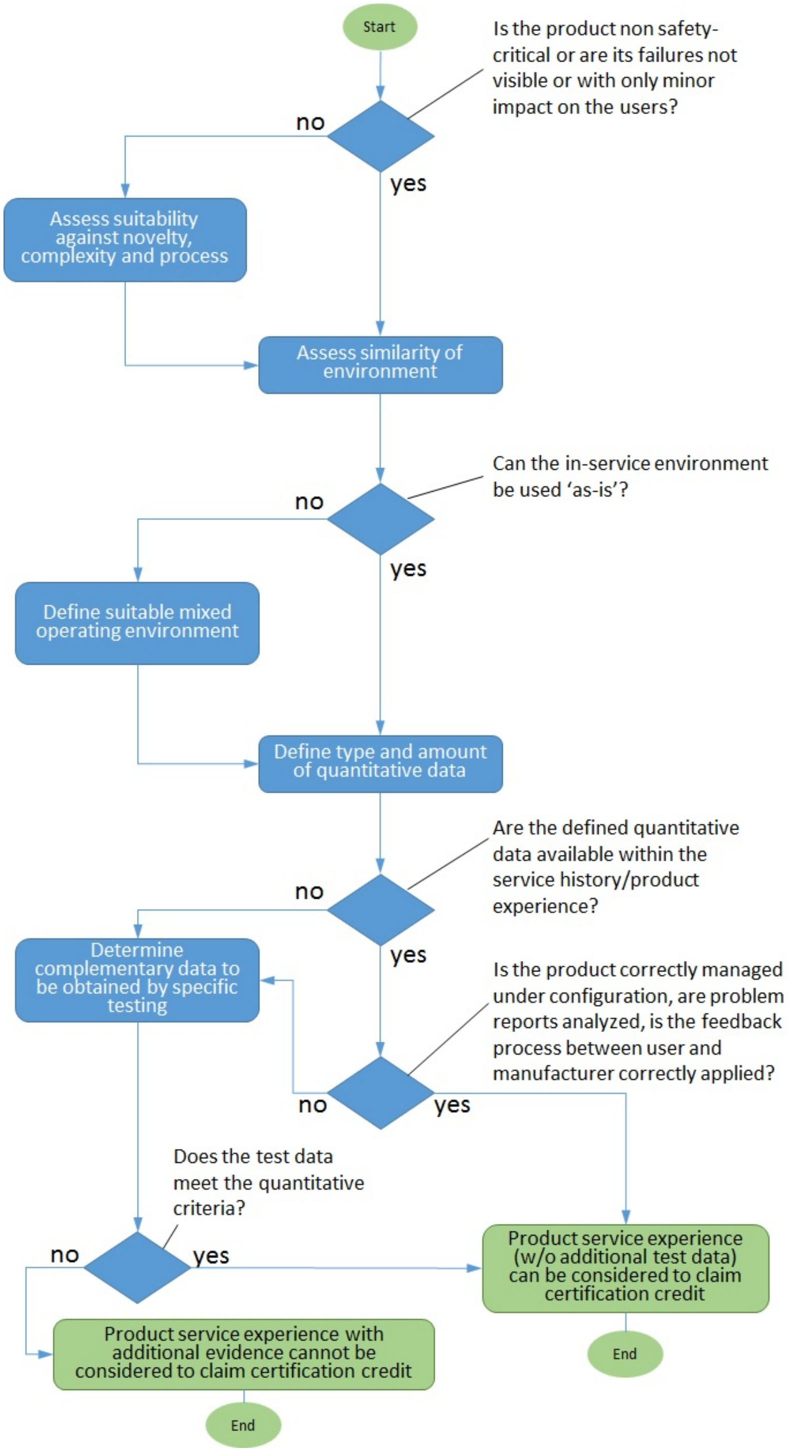
Start
Is the product non safety-critical or are its failures not visible or with only minor impact on the users?
no
yes
Assess suitability against novelty, complexity and process
Assess similarity of environment
Can the in-service environment be used 'as-is'?
no
yes
Define suitable mixed operating environment
Define type and amount of quantitative data
Are the defined quantitative data available within the service history/product experience?
no
yes
Determine complementary data to be obtained by specific testing
Is the product correctly managed under configuration, are problem reports analyzed, is the feedback process between user and manufacturer correctly applied?
no
yes
Does the test data meet the quantitative criteria?
no
yes
Product service experience (w/o additional test data) can be considered to claim certification credit
End
Product service experience with additional evidence cannot be considered to claim certification credit
End

Figure 36. Decision flow for assessing suitability

7.2.1.2 Key Processes and Activities

Another angle may be pursued that is not entirely orthogonal to the above items captured in the decision flow but is expressed in terms of activities and processes paramount to building a relevant product service experience:

- The configuration management and problem-reporting processes must be assessed and systems put in place to ensure sufficient control of the product during the in-service operations.
- The operational environments must be evaluated for similarity and differences, along with substantiation while building a service history demonstration for certification. Desirable operational environment data may be suggested to reinforce service history data collected.
- Suitable quantitative data have to be precisely defined for relevance to the certification objectives that are to be reached through the service experience alternative method.
- The event/problem-reporting process from both users and manufacturers must be defined and evaluated for efficiency. People involved in this process must be adequately trained.
- A robust data collection process between users and product manufacturers is required for efficient and detailed event feedback throughout the product life in-service. Templates for reports, regular control boards between parties, and quality assessments of the applied processes are must-haves for requesting credit-based, on-product service experience.

7.2.1.3 Using Product Service Experience for Non-Compliant or In-development Products

The suitability of product service experience is conditioned by the targeted use of the data within a certification project. The contribution of product service experience may differ if the product has existing but weak or non-compliant life-cycle data or is still in development.

For products with weak or non-compliant life-cycle data, the service experience demonstration may cover for the lack of life-cycle documentation or non-covered DO-254/DO-178C objectives that may be too tedious to achieve through regular activities (usually by reverse engineering the life-cycle data, when possible). In this case, the development process applied to the existing product has to be thoroughly evaluated for adequacy against the current certification requirements and functional requirements in the new environment:

- Identification of the standards objectives for which compliance is not achieved and activities that would correct the situation. In this case, service history (or another acceptable alternative method) would enable compliance or contribute to achieving the level of confidence commensurate with the safety objective without explicitly achieving compliance.
- Identification of the existing development and verification evidences, such as requirements and/or design/architecture development; traceability and rationales; requirement-based tests; normal operation tests; robustness tests; and endurance and random tests; etc.

For a product that is still under development, the opportunity is in anticipating the service history for future certification credit (if needed) by:

- Systematically introducing requirements relating to in-service field return during the design phase of the product. For example, by creating specific built-in test equipment requirements linked to the events and failure reporting functions. These dedicated functions will record the environmental parameters of the failure, and the operational time elapsed since the last reset, therefore providing more detailed feedback for efficient investigations to provide a solution and achieve a reliability measurement.
- Defining the correct set of data that must be fed back by the future users of the product and the FRACAS process between the manufacturer’s support/maintenance department and users.

7.2.2 Assessing Similarity of Environment

The assessment of similarity in operating environments is a key qualitative element driving both relevance and sufficiency (discussed in section 4.2). The specific items to be investigated can be organized as part of a structured analysis to evaluate the product service experience with respect to the different facets of environmental considerations. The decision flow is shown in figure 37. The following recommendations frame the context in which this analysis should be performed:

- The relevance of data for product service experience should be evaluated through a multidimensional structured analysis addressing both usage domain considerations and environmental conditions.
- The recommended multidimensional structured analysis should consider all foreseeable conditions.

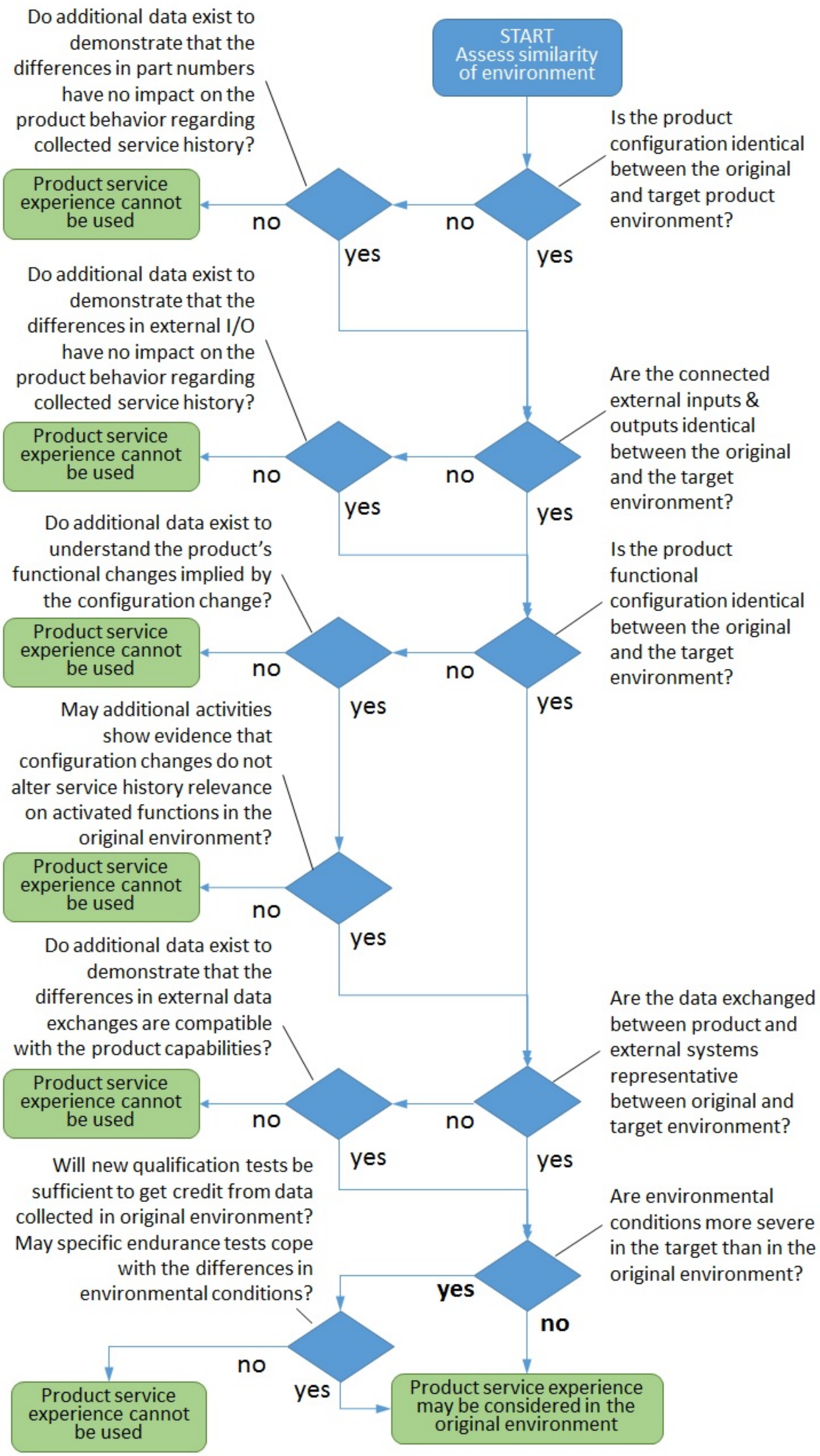
START
Assess similarity of environment
Is the product configuration identical between the original and target product environment?
no
yes
Do additional data exist to demonstrate that the differences in part numbers have no impact on the product behavior regarding collected service history?
no
yes
Product service experience cannot be used
Are the connected external inputs & outputs identical between the original and the target environment?
no
yes
Do additional data exist to demonstrate that the differences in external I/O have no impact on the product behavior regarding collected service history?
no
yes
Product service experience cannot be used
Is the product functional configuration identical between the original and the target environment?
no
yes
Do additional data exist to understand the product's functional changes implied by the configuration change?
no
yes
Product service experience cannot be used
May additional activities show evidence that configuration changes do not alter service history relevance on activated functions in the original environment?
no
yes
Product service experience cannot be used
Are the data exchanged between product and external systems representative between original and target environment?
no
yes
Do additional data exist to demonstrate that the differences in external data exchanges are compatible with the product capabilities?
no
yes
Product service experience cannot be used
Are environmental conditions more severe in the target than in the original environment?
yes
no
Will new qualification tests be sufficient to get credit from data collected in original environment? May specific endurance tests cope with the differences in environmental conditions?
no
yes
Product service experience cannot be used
Product service experience may be considered in the original environment

Figure 37. Structured analysis process for evaluating operating environment

The following recommendations pertain to the steps in the analysis shown in figure 37:

- An initial step in the analysis of environment similarity should cover the product identification and configuration.
- If the assessment of product identification and configuration does not result in the disqualification of the product service experience, key characteristics of the environment associated with domain usage to be considered in the analysis should include:

    - Physical interfaces (including relationship between the components of a hardware or software item).
    - Functional configuration.
    - Exchanged data with the product's external world.

- The change impact analysis should include considerations of product service experience, in particular whether or not the collected data remain valid or are impacted by the change.
- When assessing the physical interfaces for similarity of environment for domain usage, the analysis should support a conclusive statement on whether or not the external interfaces configuration is identical between the original domain and the target domain.
- When assessing the functional configuration for similarity of environment for domain usage, the analysis should not only evaluate the impact of activating/deactivating a function but also estimate the impact of changes in key numerical values (e.g., thresholds for filters and configuration parameters for display).
- If the target functional configuration of the product is different from the evaluated functional configuration, the following cases should be envisioned:

    - Additional information is available and can be used to support the certification credit claim (e.g., design data, user guide, installation manual, verification tests, inspection, or analysis).
    - A subset of modes common to the original and target environment can be identified so that the product service experience may be claimed for these modes.
    - Otherwise, the product service experience may not be claimed for certification credit and extensive testing is recommended to collect relevant data instead.

- When assessing the exchanged data for similarity of [domain usage] environment, the analysis should, at a minimum, consider:
    - The volume of received data.
    - The frequency at which data is received.
    - The variations in the range of data.

Regarding environmental conditions, the following recommendations are proposed:

- One approach to overcome the issue of more severe environmental conditions in the target environment would be to originally qualify a hardware item to the most severe environmental conditions that are expected in-service, in particular if reuse of the hardware is intended on multiple installations.
- Another approach to overcome the issue of varying environmental conditions would be to design products to meet multiple categories of environmental conditions—possibly using modular design or adaptable enclosures hosting core electronics. The different enclosures would be a fit for different environmental conditions.

7.2.3 Reliability Modeling

The investigation of reliability as part of this research effort was oriented toward design faults as a type of fault common to hardware and software, and toward the use of models. Other methods to improve reliability exist, but they were not investigated as part of this research (see open discussion item in section 7.5.4).

7.2.3.1 Use of Software Reliability Models

The recommended approach is to use software reliability models as a means to increase confidence in the maturity of a previously developed product that has not been certified in compliance with DO-178C or of a modified product for which DO-178C objectives may not be reachable by standard process activities. The models may allow the definition and prediction of software maturity based on a given in-service configuration. If the product is mature, it will likely be a good candidate for foreseeable development activities. It will then be evaluated against DO-178C objectives using its maturity as a support of compliance. To support this approach, the following steps are recommended:

- The determination of the usage of software functions and subfunctions should be established to segment the global software reliability into more relevant subcomponents, features, functions, or operational modes reliability.
- A faults severity and safety classification scheme should be set up in accordance with the aircraft manufacturer. The investigations conducted under this task order have considered the scheme in EASA CM-SWCEH-002 17.
- Analysis of root causes of faults is strongly recommended and should address:

    - Discarding faults that are not due to software design errors but to the design of the system itself or a hardware fault.
    - Improving faulty processes when a software design error occurs.
    - Filtering the in-service observed faults to draw more realistic software reliability models based on the system safety, availability, or continuity.

- The capture of faults from suitable data collected in-service, per section 7.2.1:
    - The correlation between the type of faults detected and the selected classification scheme.
    - The analysis of the root cause of the fault for faults classification refinement.
    - The software subcomponent, feature, function, or operational modes in which the fault occurs.
    - The evaluation of the in-service execution time for the impacted function that will help quantify the subfunction maturity.

In addition to the previous recommendations, and by referring to the software change management processes, software reliability may be improved through software modifications through the following considerations:

- The software change management process should allow a correct traceability of the modifications of the component life-cycle date, through problem reports, that will trace the faults data captured.
- A strong configuration management of the product should demonstrate traceability between the updates and software baselines over the in-service period, problem reports implemented over the successive releases, and regression test results demonstrating the correct implementation of the problem reports without adding new faults into the software components.

7.2.3.2 Selection of Statistical Model

When selecting a statistical model to fit lifetime data, several considerations play out concurrently:

- The level of details needed to address the specific objectives of the analysis.
- The available background information regarding the environmental variables and distribution of the lifetime data.
- The quality and quantity of the data available to fit the models and check their adequacy.
- The software implementation, as applicable, to perform the fit, analyze the data, and provide interpretations.

To which considerations about the type of model are added, including:

- Discrete or continuous time–The majority of methodologies for lifetime data have been developed for continuous time. The use of discretization or limit equations allow for transfer between continuous and discrete time.
- Parametric or non-parametric–If statistically fitted well, parametric models are easier to implement and can better support descriptive relationships between the model parameters and physical phenomena. Non-parametric models are less restrictive in terms of assumptions of use.

For the selection of a software reliability model, the investigations led to the following criteria pertaining to an a priori selection based on the evaluation of intrinsic qualities of the models, regardless of the observed data:

- Validity of the assumptions–The model considered for selection must be based on plausible and acceptable assumptions by software engineers.
- Applicability–The model must be able to be used in various circumstances and cases (e.g., different operational environments, different stages of the life cycle, etc.). Furthermore, the model must have a certain robustness to deviations in the assumptions.
- Capability–The model must be able to estimate the attributes relevant to the users with sufficient accuracy.
- Simplicity–The model must be conceptually simple; its theoretical foundations accessible to software engineers. Following Occam's razor principle, among competing hypotheses, the one with the fewest assumptions should be selected.
- Cost and ease-of-use–The collection of the necessary data for the estimation of these parameters should be easy and not cost-prohibitive. Underlying calculations must be easily programmable and relatively inexpensive in terms of computation time.

In addition to these a priori selection criteria, a posteriori criteria validation criteria allow the following measurements:

- The replicative capacity (i.e., the ability of the model to fit observed data).
- The predictive quality (i.e., the ability of a model to predict future failure data to meet the objectives of reliability).

To address these criteria, a multistep approach is recommended and shown in figure 38.

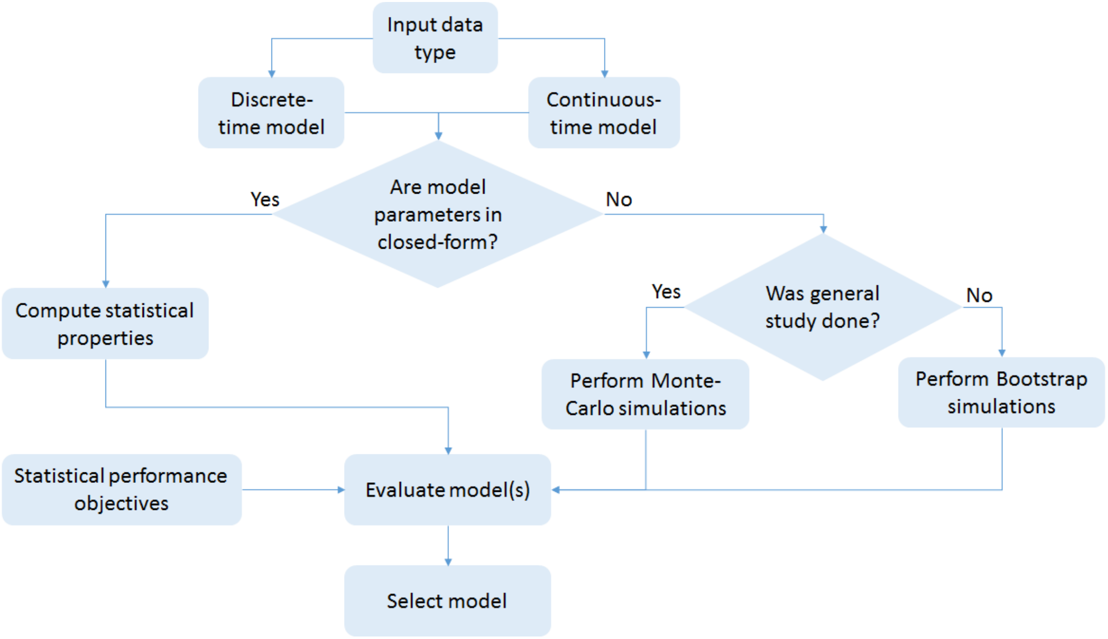
Input data type
Discrete-time model
Continuous-time model
Are model parameters in closed-form?
Yes
No
Compute statistical properties
Was general study done?
Yes
No
Perform Monte-Carlo simulations
Perform Bootstrap simulations
Statistical performance objectives
Evaluate model(s)
Select model

Figure 38. Decision flow for statistical reliability model selection

The statistical properties of reliability models in appendix C support the evaluation of the models in case a comparative assessment is required to make a final selection and against a performance objective to demonstrate compliance. The performance of the model is reflected in the quality of its reliability estimates expressed in terms of bias, dispersion, efficiency, and consistency.

The level of complexity and integration of modern avionics limits the successful identification of reliability features to the observable access provided by the data. If the data collected in service are at a component level (e.g., equipment or integrated circuit), the model will only be applicable with confidence at that level—and not at a lower level of decomposition or higher level of integration. The rules for adding estimated reliability quantities from model predictions are determined in a very limited number of cases.

7.3 RECOMMENDATIONS FOR EDAL/ELS TOP-DOWN APPROACH

7.3.1 Design Assurance Credit and Product Experience Period

Table 14 summarizes the recommendation for the product experience period needed to claim design assurance credit. See section 6 for the details on the approach.

Table 14. Summary minimum service history period per F-I DAL

F-I DAL credit	Service Period (EFH)	EDAL	PL	ELS	ELOS
F-I/DAL A	> 10^6 EFH	A	PL3/PL2*	Catastrophic	Catastrophic
F-I/DAL B	> 10^4 EFH	B	PL3/PL1*	Hazardous/ severe-major	Hazardous/ severe-major
F-I/DAL C	> 10^2 EFH	C	PL2/PL1*	Major	Major
F-I/DAL D	> 1 EFH	D	PL1/n/a	Minor	Minor
F-I/DAL E	None	E	n/a	No safety effect	No safety effect

* If PL is developed with independence

The above numbers were based on failure and system contribution statistics derived for part 25 aircraft. A similar approach needs to be developed for other parts (i.e., 23, 27, 29, and engines).

7.3.2 Problem Reporting

The current process should be modified to allow for the following classification and further refinements, so that the EDAL eligibility in table 14 can be claimed:

- CAT I problem reports cover issues having a compliance impact on the current means of compliance. These problem reports should be the ones to be considered for eligibility of service history.

- CAT II problem reports cover issues having no impact on compliance of the current means of compliance. These problem reports should be considered for upgrade or improvement on the efficiency of the current means of compliance.

7.3.3 Supporting Documentation

The following information should be provided by the applicant to justify the claim of certification credit:

- The part of the product being reused in a manner that is coherent with the PL used to claim the EDAL.
- The domain in which in-service data were captured (e.g., public domain, railways, automobile, or military).
- A minimal set of relevant data, identified through a predefined process, which should be the minimum allowed skeleton for service-history eligibility.
- A description of the context of data collection, including the quality process used to capture these data.
- A description of the process used to monitor the continuing performance of the item in service (e.g., system, LRU, or component).
- A description of the problem correction process throughout the data collection period in a coherent manner with the EDAL being claimed.
- The typology of recorded information, including the data capture frequency and related issue reports.

7.4 CONCLUSION

7.4.1 Data Collection

Unless the requirements for service history or product experience are (or were) integrated in the product development phase, the data that can be collected while in-service is likely to be limited compared with the desired data set. The available collected data and the level of details in the issue reports are to be balanced against the guidance in the standards for claiming certification credit. The data collection process involves multiple actors. These actors focus on distinct objectives that may compete against each other when considering the data analysis and error correction processes. Therefore, both the number of actors involved in the data collection process and the variability in their objectives may impact the usability of the collected data. Recommendations for minimum information content to be collected are developed, though the ultimate decision is that of the OEM.

7.4.2 Using Qualitative and Quantitative Criteria for Software and AEH

Using the proposed bottom-up approach to claim certification credit based on software service history/product service experience, the determination of what constitutes suitable data needs to be addressed both from a qualitative and quantitative point of view. The qualitative assessment allows the determination of what type of quantitative data will be suitable. Suitability is conditioned on such qualitative attributes as criticality (assurance level), complexity, level of innovation, and impact on the user. Furthermore, suitability may be constrained by operating environment conditions and processes associated with the product (software/hardware). The determination of the similarities and differences between the in-service and target environments is crucial and should be performed via an organized and comprehensive analysis of both the usage domain and the environmental conditions for all foreseeable conditions. The performance of this analysis may be impeded by three factors: 1) the level of control on both the in-service and target environment necessary to evaluate their differences, 2) amount and quality of available information to obtain a sufficient level of confidence in the data collection process, and 3) absence of objective acceptability criteria.

For each of the above elements, the applicant may be guided through structured question-based decision flows wherein a negative answer should lead to reconsidering the suitability of the product service experience data for the purpose of claiming certification credit. These decision flows should be integrated in guidance documents for the applicant and primarily apply to software and hardware levels.

The quantitative criteria include the selection of meaningful measurement units with respect to the operating environment and the provision of reliability estimates. Reliability can be considered an application domain of lifetime data, wherein the data are modeled in terms of random processes to capture the indeterminacy in the evolution from the initial condition to the next state. For aircraft and complex airborne systems, this evolution also includes alternatives, design trade-offs, changes in operational environments, etc. Numerous models exist in the literature to describe such data, so much so that the difficulty is rather to select the most adequate model and determine the quality of its estimates.

The selection of a suitable model follows a decision flow that first considers the problem at hand (e.g., the objective of claiming certification credit, the quality and quantity of available data); then the model’s intrinsic properties, regardless of the actual data they would apply to (e.g., applicability of model’s assumptions, achievable accuracy, simplicity of implementation); and, finally, the model properties applied to the available data (e.g., quality of fit, confidence in prediction). Several quality-of-fit criteria can be used to describe the reliability estimate obtained through modeling.

7.4.3 Using EDAL and ELS at System Level

The use of system theory to establish an EDAL based on a PL allows for eliminating most issues associated with the lack of detailed information and reliability modeling issues described in the bottom-up approach. Failure statistics for part 25 aircraft support the proposition of a minimum service history period as a function of EDAL and ELS to support a claim of certification credit. This approach covers all assurance levels and is extendable to other parts, provided similar statistics exist. More research is needed on the justification of the EDAL classification and its practical implementation and modifications to the problem-reporting process.

7.4.4 Combined Decision Table

The goal of table 15 is to condense the findings and the recommendations into the initial framing format of the research, in an effort to provide the reader with a concentrated takeaway.

Table 15. Condensed list of findings and recommendations

Questions	Bottom-Up Approach	Top-Down Approach	Limitations	Is More Research Needed?
Use	All decision trees lead to a positive answer.	When ELOS can be provided.	Bottom-up approach implemented but currently limited to DAL D AEH.	Yes, for both bottom-up approach quantitative criteria and top-down minimum history recommendation and EDAL.
Relevance	Similarity of environment decision tree is positive.	History data support recommended PL and system description.	Deficiencies in data collection and problem reporting.	Yes, practical implementation of PL and system description from collected data, mitigation of deficiencies in processes.
Sufficiency	Based on quality-of-fit of reliability model to in-service data.	Based on allocation of safety objectives to system and EDAL.	Direct link between quality-of-fit of model and minimum flight hours of data is missing.	Yes, for quality-of-fit figures (bottom-up approach) and for minimum history period for parts 23, 27, 29, and 33 for top-down approach.
Problem reporting	Impacts the filtering of data to be included in the modeling. May lead to overly optimistic or pessimistic reliability estimates.	Impacts the decision on the eligibility of the data.	Lack of detailed classification.	Yes, detailed classification of problem reports to support either approach using criteria such as user impact or compliance impact.
Documentation	Additional substantiation to support the decision tree branch points.	Additional substantiation of the EDAL.	N/A	Yes, process to develop the additional documentation, minimum information and format required.
Credit type	Partial. Mainly increased confidence.	Partial or full per applicable safety goals mapping to minimum PL.	Ability to collect the underlying information.	Yes, practical implementation of the top-down approach.

7.5 OPEN DISCUSSION ITEMS

This section consolidates discussion points that were raised in the intermediate white papers and their reviews. It is primarily intended to form the core of proposed further research on the topic as part of the final report.

7.5.1 Fault Classification Scheme

The determination of suitability as it relates to the demonstration of capture, identification, and correction of faults according to their impact supposes the existence of an agreed fault classification scheme. The methodology and recommendations within this research used the fault classification scheme derived from open problem report types in [17], which comprise the performing organization's background knowledge and current practice. Moreover, the performing organization's worldwide repair centers use a simpler scheme that is based on qualifying problems as major or minor.

EASA's classification [17] was reviewed as potentially too simplistic to be useful for establishing modeled proof of design and safety assurance. However, fault classification is a strong requirement for several criteria of relevance and problem reporting for software service history and AEH service experience. Depending on how the faults observed in service are classified and associated to root causes, the measure of reliability may be more pessimistic than needed. To the authors' knowledge, no other classification scheme for observed faults formally exists that ties back to the impact of the fault (fault classification schemes do exist that trace faults to the phase in which they were introduced, such as orthogonal defect classification).

The recommendation would therefore be to investigate the requirements for key features of a fault classification scheme for use in safety assurance and to perform a gap analysis with schemes currently applied. The potential worst impact of defining a scheme that cannot be traced in some way to current practices is that it would invalidate service data. In addition, current tools used for the collection of problem reports and associated data should be reviewed with regard to coverage of the defined key features of a desired fault classification scheme.

7.5.2 Failure Rate for an Electronic Product

The issue at hand is the development of smooth transitions for reliability estimates at various levels—from components of a microcircuit to the equipment. Apart from considerations of complexity of the model to be commensurate with the underlying data it represents and the phenomenon it modeled, only scarce guidelines can be found and often times with little explanation to the potential user. This section provides examples at each end of the spectrum.

The FIDES methodology [30] proposes that the global failure rate of an electronic product (usually equipment, but excluding software) be the numerical sum of the failure rates for each of its constituents:

$$\lambda_{prod} = \sum_{components} \lambda_{components} + \sum_{PCB} \lambda_{PCB} + \sum_{COTSboards} \lambda_{COTS} + \sum_{oth\_subassemblies} \lambda_{oth\_SA} \tag{9}$$

In the ongoing work by the AVSI under project AFE 83, a reliability modeling tool has been developed to support users in determining reliability at microcircuit level [55]. The tool considers both random failure lifetime (flat portion of bathtub) and wear-out lifetime, and includes models for various failure modes that have been used in literature/developed in previous AVSI projects. Table 16 provides an overview.

Table 16. AFE 83 recommended models per failure mode

Failure Mode	Power Law	Exponential	Inverse Exponential
Time-dependent dielectric breakdown	Default	Selectable	Selectable
Negative bias temperature instability	Default	Selectable	N/A
Electromigration	Default	N/A	N/A
Hot carrier injection	Selectable	N/A	Default

The reliability modeling tool allows for the user to retain each individual failure rate or obtain a combined rate using four approaches. The selection of the approach is dependent on the test setup and user objectives:

- Failure rates are numerically added–This approach could be argued if a single experiment were providing the results, or if a conservative (bounding) total failure rate were sought.
- Failure rates are equally weighted–This approach could be argued if the user had no information on which failure mode dominates or when no failure occurred during the test time. Each of the four failure rates is weighted as a 25% contribution to the total failure rate.
- Failure rates are proportionally weighted–This approach is the default for the tool. Each failure rate is weighted based on the proportion of the total acceleration factor, so that failure modes poorly accelerated by the experiment setup are less dominating as part of the total failure rate.
- Failure rates are custom-weighted–This approach allows the user to adjust for the supplier's specific information or support the decision to ignore failure modes that might not have been well-accelerated by the experiment.

The approach by AVSI AFE 83 seems richer in the sense that it includes the numerical sum approach as a bounding approach while introducing knowledge of failure mode dynamics to relatively weighted failure rates. Further research and substantiation using historical test and in-service data at levels above microcircuit would allow for the description of conditions for the selection of a combining approach against another.

7.5.3 Modeling and Quality of Fit Considerations

The issue at hand is twofold: 1) determining applicable statistical tests for quality-of-fit of the models presented in this report and 2) substantiating the selection of value threshold. That is, proposing statistical metrics for the validation of these models.

As illustrated in the examples in section 5.3, if the quantitative reliability is based on modeling, and the model is fit to historical data, then the following should be considered at a minimum:

- The model needs to be convergent:
    - Either convergence is one of its statistical properties (e.g., from literature or standard).
    - There is sufficient historical data to show the estimated reliability metric remains within acceptable bounds (e.g., the estimator error is a strictly decreasing function, or the estimator error is 2% or lower for 95% of the data).

- The model trends need to be explained, especially when showing later in time:
    - They can be the sign of premature aging related to technology scaling.
    - They can be the sign that the historical data used to fit the model are not sufficient.

Literature on reliability provides statistical methods to compute confidence intervals, and some models have closed forms for confidence bounds. What seems to remain open is the characterization of an objective confidence level that could be related to the DAL at equipment level or safety objective associated with the failure condition that the model is capturing. With models typically following exponential shapes for reliability growth and wear-out lifetimes, an argument for how much data are needed (or how much time history is required) could be produced. Discussing the impact of early wear-out on modeling and predict MTBF would require more research.

Finally, most of the data accessible in the literature relate to reliability modeling for hardware. Very little exists for software and even less for systems integrating hardware and software unless the level of abstraction is at system failure in the MTBF sense and not for a specific failure mode. The conditions under which modeling would provide usable predicted reliability would be of interest to inform on where and when to stop robustness tests on software in the reliability growth lifetime (e.g., when software reaches an acceptable level of failure rate prior to in-service operation).

7.5.4 Methods to Improve Software Reliability

As noted in the FAA’s software service history report [4], the current position is that methods for estimating probabilities of software errors have not yet provided a level of confidence acceptable for software assurance purposes. Since the publication of the report and DO-178C document, work on software reliability has improved the level of confidence. However, the multiplicity of available models and absence of quantified performance objectives contribute to the confidence issue remaining open and added the issue of guidelines on selecting an adequate reliability method.

This research, though recognizing the existence of other methods for improving software reliability, focused on models because their characteristics and method for selection can be, to a great extent, common between hardware reliability and software reliability (though the reliability of software and hardware is different). This report did, however, touch upon the three categories in which all methods and techniques pertaining to improving software reliability fall:

1. Fault avoidance–Process-oriented methods aiming at preventing the introduction of faults during the software development phase.
2. Fault detection–Product-oriented methods aiming at detecting faults once the software is developed.
3. Fault tolerance–Safety-critical and availability focused techniques aiming at providing a controlled response of the product in the event of previously uncovered faults.

Formal methods that were not covered by this research and apply toward the goal of improving software reliability include deductive verification and model checking. Methods applied during software execution (e.g., semantic analysis) and methods for software fault-tolerance or resilience have not been investigated. Software hardening and the establishment of rules for quantification of complexity have not been discussed.

In addition, processes compliant with DO-178C guidelines will not prevent system specification errors, design errors, or lack of containment for physical faults. Approaches to correct this omission in the currently applicable standard were not discussed as part of this research but should be addressed. Similarly, approaches should be suggested for future research to address process improvements/additional verification processes to specifically reduce the occurrence of design errors not captured by the verification and validation process.

One of the main assumptions in discussing software reliability versus hardware reliability has been that software does not show wear-out phenomena. By extension, the impact of software aging through fault-induced memory leaks and software rejuvenation on the validity of software service history have not been discussed.

7.5.5 Comparison with FAA Handbook on Selection of Microprocessors

The FAA's handbook on the selection and evaluation of microprocessors [26] endorsed the concept of a multilayer safety net against which the selection of COTS microprocessors may be justified. The questions to be answered for first time use are in complete alignment with the investigations performed under this research contract:

- How long has the microprocessor been fielded? This question is covered under items pertaining to the sufficiency of the service experience.
- What has it been used for? This question is discussed within the theme of evaluating similarity of operating environment, in particular regarding domain usage.
- Is there substantiated service history that can be evaluated? This question covers relevance, sufficiency, and considerations of problem reporting within this research.

The consideration of software monitoring features (e.g., built-in test) is part of the approach. The specification of these monitoring features is key in preventing and capturing faults.

The proposed approach to use qualitative and quantitative criteria to evaluate suitability of the data, maturity (stability), and similarity of environment should echo the particulars for COTS microprocessors in the FAA’s handbook. However, no detailed comparative analysis has been performed as part of this research.

A proposal to be investigated would be to decompose the microprocessor into a general purpose component for which service history could be used to obtain partial certification credit and a specific use component that would be tested against standard assurance processes. A comparison point in the approach could be the work by Michael Holloway (NASA) on trying to back out a safety case [56].

8. GLOSSARY

Key words are defined differently in the various reference standards. Furthermore, different domains use different appellations to describe the lifetime data of software or hardware collected after entry into service. This glossary collected the various terms and corresponding acceptation used in this report and their source, as applicable (see table 17).

Table 17. Glossary of service history related definitions

Term or Concept	Definition
Abnormal Test Case	Type of test case that reflects an unacceptable, abnormal, or unexpected condition or data to demonstrate that the requirement is only achieved under the desired conditions. These test cases are also referred to as negative test cases. In the context of software, DO-178C refers to these test cases as robustness test cases.
Change Impact Analysis	Identifying the potential consequences of a change in a design or estimating what needs to be modified to accomplish a change. Other definitions focus on the identification of risks associated with the change and the estimation of consequences on resources, effort, and schedule. Both the risk and the design aspects are critical to the performance of impact analysis within the change management processes. Impact analysis techniques fall into three categories: traceability, dependency, and experiential.
Commercial off-the-shelf	
RTCA DO-254	Component, integrated circuit, or subsystem developed by a supplier for multiple customers whose design and configuration are controlled by the supplier's, or an industry, specification.
RTCA DO-297	Commercially available applications sold by vendors through public catalog listings.
RTCA DO-178C	Commercially available applications sold by vendors through public catalog listings. COTS software is not intended to be customized or enhanced. Contract negotiated software developed for a specific application is not COTS software.
RTCA DO-278A	Software under consideration for use in a CNS/ATM system that may have no, or only partial, evidence of compliance to this document [RTCA/DO-278A], sections 4–9 objectives.
EUROCAE/ED-153	COTS software encompasses a wide range of software, including purchased software, non-developmental items, and software previously developed without consideration of ED-153. This software may or may not have been approved through other "approval processes." Partial data or no data may be available as evidence of objectives of ANS developmental process.

Table 17. Glossary of service history related definitions (continued)

FDA, UCM073778	A generally available software component used by a medical device manufacturer for which the manufacturer cannot claim complete software life-cycle control.
IEEE 24765:2010	1) Software defined by a market-driven need, commercially available, and whose fitness for use has been demonstrated by a broad spectrum of commercial users. 2) Software product available for purchase and use without the need to conduct development activities. 3) An item that a supplier offers to several acquirers for general use.
Complexity	
RTCA DO-297 EUROCAE/ED-124	An attribute of systems or items that makes their design/operation difficult to comprehend.
ARP-4754A	An attribute of functions, systems, or items that makes their operation, failure modes, or failure effects difficult to comprehend without the aid of analytical methods.
IEEE 24765:2010	1) The degree to which a system's design or code is difficult to understand because of numerous components or relationships among components. 2) Pertaining to any of a set of structure-based metrics that measure the attribute in (1). 3) The degree to which a system or component has a design or implementation that is difficult to understand and verify.
System theory [57]	A system whose overall behavior is characterized by reduced predictability.
Deviation from rules EASA/SWCEH-001	A non-conformity of the development process with the plans, development standards, and applicable certification review items. In the particular case where a non-conformity with the plans or development standards is intentional, and the plans or development standards are planned to be modified accordingly, such a non-conformity might not be recorded as a problem but instead be identified and justified in the compliance status of the Hardware Accomplishment Summary.

Table 17. Glossary of service history related definitions (continued)

Design phase	
From high-level requirement capture to coding:	
IEEE 24765:2010	1) The process of defining the architecture, components, interfaces, and other characteristics of a system or component. 2) The result of the process in (1). 3) The process of defining the software architecture, components, modules, interfaces, and data for a software system to satisfy specified requirements. 4) The process of conceiving, inventing, or contriving a scheme for turning a computer program specification into an operational program. 5) Activity that links requirements analysis to coding and debugging. 6) Stage of documentation development that is concerned with determining what documentation will be provided in a product and what the nature of the documentation will be.
Environment IEEE 24765:2010	1) Anything affecting a subject system or affected by a subject system through interactions with it, or anything sharing an interpretation of interactions with a subject system. 2) The configuration(s) of hardware and software in which the software operates. 3) The circumstances, objects, and conditions that surround a system to be built. 4) The circumstances, objects, and conditions that will influence the completed system. 5) A concept space.
Environmental Conditions (foreseeable conditions) FAA/AC 11-25	Foreseeable conditions means the full environment in which the display or the display system is assumed to operate, given its intended function. This includes operating in normal, non-normal, and emergency conditions. Foreseeable environmental conditions are used in 14 CFR 25.1309(e).
Error	
RTCA DO-178C	A mistake in requirements, design, or code.
EASA/SWCEH-002	With regard to software, a mistake in requirements, design, or code.
EASA/SWCEH-001	A mistake in requirements, design, or implementation.
IEEE 24765:2010	1) A human action that produces an incorrect result, such as software containing a fault. 2) An incorrect step, process, or data definition. 3) An incorrect result. 4) The difference between a computed, observed, or measured value or condition and the true, specified, or theoretically correct value or condition.

Table 17. Glossary of service history related definitions (continued)

Failure	
RTCA DO-178C	The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered.
EASA/SWCEH-002	The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered. But a fault may also remain hidden at system level and have no operational consequences.
RTCA DO-254	The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered. A failure is a manifestation of a fault at system level.
EASA/SWCEH-001	The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered. But a fault may also remain hidden at system level and have no operational consequences.
IEEE 24765:2010	1) Termination of the ability of a product to perform a required function or its inability to perform within previously specified limits. 2) An event in which a system or system component does not perform a required function within specified limits.
Failure (catalectic) ISO/TR 12489:2013	This term has been introduced by analogy with the catalectic verses (i.e., a verse with seven foots instead of eight), which stop abruptly. 1) Then, a catalectic failure occurs without warning and is more or less impossible to forecast by examining the item. It is the contrary of failures occurring progressively and incompletely. 2) Catalectic failures characterize simple components with constant failure rates (exponential law): They remain permanently “as good as new” until they fail suddenly, completely, and without warning. Most of the probabilistic models used in reliability engineering are based on catalectic failures of the individual component of the system under study (e.g., Markovian approach).
Failure Condition	
RTCA DO-178C EASA/SWCEH-002	The effect on the aircraft and its occupants, both direct and consequential, caused or contributed to by one or more failures, considering relevant adverse operational and environmental conditions. A failure condition is classified according to the severity of its effect, as defined in advisory material issued by the certification authority.
RTCA DO-254 EASA/SWCEH-001	The effect on the aircraft and its occupants both direct and consequential caused or contributed to by one or more failures, considering relevant adverse operational and environmental conditions. A failure condition is classified according to the severity of its effect, as defined in FAA AC 25.1309 or AMC 25.1309.

Table 17. Glossary of service history related definitions (continued)

Failure Intensity	The failure intensity is defined for repairable systems or components as the anticipated number of times the repairable system or component will fail in a specified time period, given that it was GAN at time zero and functioning at time *t*. It is typically expressed as number of failures per hour, though other units can be used (e.g., miles or revolutions). For electronic devices and, in particular, semiconductors, failure in time is typically used to express the number of failures in one billion hours of operations.
Failure Rate	The failure rate is defined for non-repairable systems or components as the anticipated frequency at which the non-repairable engineering system or component will fail. It is typically expressed using a time unit in hours, though other units can be used (e.g., miles or revolutions).
Fault	
RTCA DO-178C EASA/SWCEH-002	A manifestation of an error in software. A fault, if it occurs, may cause a failure.
EASA/SWCEH-001	1) A manifestation of a flaw in hardware due to an error or random event. A fault, if it occurs, may cause a failure. 2) An undesired anomaly in a device.
IEEE 24765:2010	1) A manifestation of an error in software. 2) An incorrect step, process, or data definition in a computer program. 3) A defect in a hardware device or component.
In-Service Hour RTCA DO-278A	Use of software for one hour in a controlled environment functionally equivalent to the target environment
Level of Confidence (rail applications)	
XX CFR 236	A high degree of confidence is characterized by: “as applied to the highest level of aggregation, means there exists credible safety analysis supporting the conclusion that the likelihood of the proposed condition associated with the new product being less safe than the previous condition is very small” [58].
XX CFR 229	A high degree of confidence is characterized by: “as applied to the highest level of aggregation, means there exists credible safety analysis supporting the conclusion that the risks associated with the product have been adequately mitigated” [59].
Life-Limited Component	A component for which the wear-out failure mechanisms can appear before the end of service.

Table 17. Glossary of service history related definitions (continued)

Life Test	Process of testing a product under specific environmental conditions (e.g., stress, strain, temperatures, voltage, vibration rate, and pressure) and observing the time to failure. The most frequent type of life test is accelerated life testing, in which the test conditions are in excess of the normal service parameters to uncover faults and failure modes in a short amount of time.
Maturity	Maturity is most often used in a financial/marketing context to indicate the period in a product's life in which sales peak. In the context of reliability for this report, maturity is associated with stability of the product, measured by the rate of discovery of failures and errors in the product in service. For certain types of failures, the word maturity is used to denote that the failure intensity function (bathtub curve) of the product is flat or, that is, that the discovery of failures is constant over time (see also Failure (catalectic)).
Memoryless	The qualifier of "memoryless" is given to probability distributions or stochastic processes for which the past has no bearing on the future behavior. Every instant is equivalent to the beginning of a new random period, regardless of how much time has elapsed. In this context, the probability of a failure to occur is the same regardless of how long the system has already been running.
Novelty SAE/ARP4761	Applicable to systems using new technology and systems using a conventional technology not previously used in connection with the particular function in question. Note that in this report, novelty has been elicited by "level of innovation."
Open problem report EASA/SWCEH-001	A problem that has not been corrected at the time the AEH is presented for approval.
Parametric model	In statistics, a parametric model is a family of distributions that can be described using a finite number of parameters. In general, a parametric model is entirely defined in terms of parameters. When a model cannot be entirely defined by a finite set of parameters, it is said to be non-parametric.
Previously Developed Software	
RTCA DO-178C, 278A	Software already developed for use. This encompasses a wide range of software, including COTS software through software developed to previous or current software guidance.

Table 17. Glossary of service history related definitions (continued)

EUROCAE/ED-153	In the context of ED-153, the term "Previously Development Software" may be used in lieu of COTS software. The definition of such software in ED-153 encompasses a wide range of software, including purchased software, non-developmental items, and software previously developed without consideration of ED-153 (see the definition of COTS for ED-153).
IEEE 24765:2010	Software that has been produced prior to or independent of the project for which the plan is prepared, including software that is obtained or purchased from outside sources.
Product	
IEEE 24765:2010	1) An artifact that is produced, is quantifiable, and can be either an end item in itself or a component item. 2) Complete set of software and documentation. 3) Output of the software development activities. 4) Result of a process.
SAE ARP4761	An item generated in response to a defined set of requirements.
Product Log	
Product Service Experience RTCA DO-254	A period of time during which the hardware is operated within a known environment and during which successive failures are recorded. Service experience relates to data collected from any previous or current usage of the component (see section 11.3).
Product Service History	
RTCA/DO-178C	A contiguous period of time during which the software is operated within a known environment, and during which successive failures are recorded.
RTCA/DO-297	A contiguous period of time during which an aircraft, product, or part thereof is operated within a known environment and during which failures are recorded.
FAA System Safety Handbook	Historical data generated by activities at the interface between the supplier and the customer and by supplier internal activities to meet the customer needs regarding the quality, reliability, and safety trends of the product or service.
Qualitative AC 25.1309-1A	Analytical processes that assess system and airplane safety in a subjective, non-numerical manner.
Quantitative AC 25.1309-1A	Analytical processes that apply mathematical methods to assess system and airplane safety.
Reliability	
SAE ARP4761	The probability that an item will perform a required function under specified conditions, without failure, for a specified period of time.

Table 17. Glossary of service history related definitions (continued)

RTCA DO-254	The probability that an item will perform its intended function for a specified interval under stated conditions.
RTCA DO-297	1) The quantitative attribute and measure of dependability with regard to the continuity of the service. In a quantified way, it is the conditional probability that the system or component thereof has survived in a specified environment until the time *t*, given that it was operational at time 0. A frequently used estimator associated with this measure is the mean time to first failure. 2) The probability that a component will perform a required function under specified conditions, without failure, for a specified period of time.
IEEE 24765:2010	1) The ability of a system or component to perform its required functions under stated conditions for a specified period of time. 2) The capability of the software product to maintain a specified level of performance when used under specific conditions. Reliability can, in theory, be measured at levels ranging from component to system. For example, reliability data can be obtained for the devices in the system (e.g., memory chip); the integrated circuits or boards; the equipment; and the system. For software, reliability information can be obtained at the function and subfunction levels.
Re-use Items MIL-STD-882E	Items previously developed under another program or for a separate application that are used in a program.
Robustness	
RTCA DO-178C	The extent to which software can continue to operate correctly despite abnormal inputs and conditions.
IEEE 24765:2010	The degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions.
Safety Critical	
MIL-STD-882E	A term applied to a condition, event, operation, process, or item whose mishap severity consequence is either catastrophic or critical. Note that mishap in the context of MIL-STD-882E equates to accident for the FAA community (see Safety System Team below).
FAA System Safety Handbook	All interactions, elements, components, subsystems, functions, processes, and interfaces within the system that can affect a predetermined level of risk.
Safety System Team	A formally chartered group of persons, representing different engineering specialties relevant to the system under study, who are organized to assist the system safety manager in achieving the system safety objectives.

Table 17. Glossary of service history related definitions (continued)

Service Experience RTCA DO-178C, RTCA DO-278A	Intervals of time during which the software is operated within a known relevant and controlled environment, during which successive failures are recorded.
Similarity RTCA DO-254	Applicable to systems comparable in characteristics and usage to systems used on an airplane previously certificated by the applicant. It is further assumed that there are no parts of the subject system more at risk due to environment or installation and that operational stresses are no more severe than on the analogous system.
Software Aging	Software aging covers the deterioration of operating system's resources, data corruption, and numerical error accumulation [60].
Software Rejuvenation	Software rejuvenation is a proactive fault management technique aimed at preventing performance degradation and other failures associated with software aging. Several more or less complex techniques exist to achieve rejuvenation—ranging from a simple reboot to cleaning up the system's internal state to prevent the occurrence of a more severe crash [60].
Specific Component	A component for which no reliability data are available or no standard model is applicable.
Standard Component	A component for which the reliability can be predicted using standards such as MIL-HDBK-217, FIDES methodology, or equivalent.
User	
EASA/SWCEH-002	An airline/operators of the software
IEEE 24765:2010	1) Person who performs one or more tasks with software. 2) Person who interacts with the product. 3) Individual or organization who uses a software-intensive system in daily work activities or recreational pursuits. 4) Individual or group that benefits from a system during its utilization. 5) Any person or thing that communicates or interacts with the software at any time. 6) A person (or instance) who uses the functions of a computer-based software via a terminal by submitting tasks and receiving the computed results. 7) The person who derives engineering value through interaction with a computer-aided software engineering tool.

9. REFERENCES

1. "Software Considerations in Airborne Systems and Equipment Certification," RTCA DO-178C, December 2011.

2. "Design Assurance Guidance for Airborne Electronic Hardware," RTCA DO-254, April 2000.

3. "Software Service History Handbook," FAA report DOT/FAA/AR-01/116, January 2002.

4. "Software Service History Report," FAA report DOT/FAA/AR-01/125, January 2002.

5. "Airborne Software Assurance," FAA AC 20-115C, July 19, 2013.

6. EASA, AMC 20-115C, Software Considerations for Certification of Airborne Systems and Equipment.

7. "Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems," RTCA DO-278A, December 2011.

8. "Guidelines for ANS Software Safety Assurance," EUROCAE Report ED-153, August 2009.

9. "Supporting Information for RTCA DO-178C and RTCA/DO-278A," RTCA DO-248C, December 2011.

10. "Software Reusable Components," FAA AC 20-148, December 7, 2004.

11. "Clarification on the Use of RTCA Document RTCA/DO-254 and EUROCAE Document ED-80, Design Assurance Guidance for Airborne Electronic Hardware," CAST Position Paper #27.

12. "Use of COTS Graphical Processor (CGP) in Airborne Display Systems," CAST Position Paper #29.

13. "Technical Clarifications Identified for RTCA DO-254/EUROCAE ED-80," CAST Position Paper #31 Revision 4.

14. "Simple and Complex Electronic Hardware Approval Guidance," FAA Order 8110.105 Change 1.

15. FAA Order 8110.49. (2003). "Software Approval Guidelines (Change 1)."

16. "Certification Memorandum on Development Assurance of Airborne Electronic Hardware," EASA Report CM-SWCEH-001, March 2012.

17. "Certification Memorandum on Software Aspects of Certification," EASA report CM-SWCEH-002, March 2012.

18. "Nuclear Power Plants—I&C Systems Important to Safety—Software Aspects for Computer Based Performing Category A Functions," IEC standard 60880.

19. "Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems. Part 3—Software Requirements," IEC standard 61508.

20. “Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software Use in Medical Devices,” FDA report.

21. Electronic Code of Federal Regulations (e-CFR), available at www.ecfr.gov (accessed on 01/11/2016).

22. DeWalt, M. and McCormick, G.F., “When Is An Objective Not An Objective In Technology Independent Assurance Method,” *33rd Digital Avionics Systems Conference (DASC)*, Colorado Springs, Colorado, 2014.

23. Meeker, W.Q. and Escobar, L.A., *Statistical Methods for Reliability Data*, Wiley, 1998.

24. “Failure Reporting, Analysis and Corrective Action System,” DoD report MIL-STD-2155, July 1985.

25. Marquez, A.C., *The Maintenance Management Framework: Models and Methods for Complex Systems Maintenance*, Springer Series in Reliability Engineering, 2007.

26. “Handbook for the Selection and Evaluation of Microprocessors for Airborne Systems,” FAA report DOT/FAA/AR-11/2, February 2011.

27. “Environmental Conditions and Test Procedures for Airborne Equipment,” RTCA DO-160G, December 2010.

28. “Electronic Flight Deck Displays,” FAA AC 25-11A, June 2007.

29. “Reliability Prediction of Electronic Equipment,” DoD report MIL-HDBK-217F, January 1990.

30. FIDES, methodology and documents downloadable from fides-reliability.org (accessed on 01/11/16).

31. Schaefer, I.R., Rabiser, D., and Clarke, L., et al., “Software Diversity: State of the Art and Perspectives,” *International Journal on Software Tools for Technology Transfer*, Springer Verlag (Germany), 2012, 14 (5), pp. 477–495.

32. Lawless, J.F., *Statistical Models and Methods for Lifetime Data*, Wiley, 2003.

33. Rigdon, S.E. and Basu, A.P., *Statistical Methods for the Reliability of Repairable Systems*, Wiley, 2000.

34. Singpurwalla, N.D. and Wilson, S.P., *Statistical Methods in Software Reliability: Reliability and Risk*, Springer, 1999.

35. Regis, D., Berthon, J., Hubert, G., and Gatti, M., “DSM Reliability Concerns—Impact on Safety Assessment,” Proceedings of the 33rd Digital Avionics Systems Conference (DASC), 2014.

36. Regis, D., Hubert, G., Bayle, F., and Gatti, M., “IC Components Reliability Concerns for Avionics End-Users,” Proceedings of the 32nd Digital Avionics Systems Conference (DASC), 2013.

37. Crow, L.H., “Planning a Reliability Growth Program Utilizing Historical Data,” Proceedings of the Annual Reliability and Maintainability Symposium (RAMS), 2011.

38. “Handbook, Reliability Growth Management,” Department of Defense Report MIL-HDBK-189C, June 2011.

39. von Bertalanffy, L., *General System Theory: Foundations, Development, Applications*, George Braziller, 1968.

40. Ashby, W.R., *An Introduction to Cybernetics*, Chapman & Hall Ltd., London, 1957, Chapter 12, “The Error-Controlled Regulator.”

41. Wiener, N., “3.5 Feedback and Oscillation,” in *Cybernetics: Or Control and Communication in the Animal and the Machine*, Hermann & Cie (Paris) and The MIT Press, Cambridge, MA, 2nd revised ed., 1961.

42. Russell, B., “On Denoting,” in *Mind, New Series*, Vol. 14, No. 56, pp. 479–493, October 1905, available at http://bactra.org/Russell/denoting (accessed on 01/11/16).

43. Ashby, W.R., *An Introduction to Cybernetics*, Chapman & Hall Ltd., London, 1957, Chapter 7, “Quantity of Variety.”

44. Leveson, N., “3. System Theory and its Relationship to Safety” and “4.3. Process Models” in *Engineering a Safer World: System Thinking Applied to Safety*, The MIT Press, Cambridge, MA, 2011.

45. Ashby, W.R., *An Introduction to Cybernetics*, Chapman & Hall Ltd., London, 1957, Chapter 2, “Change.”

46. “Guidelines for Development of Civil Aircraft and Systems,” SAE report ARP4754A, 2010.

47. Leveson, N., “The Use of Safety Cases in Certification and Regulation,” *Journal of System Safety*, Vol. 47, No. 6, November–December 2011.

48. Ashby, W.R., *An Introduction to Cybernetics*, Chapman & Hall Ltd., London, 1957, Chapter 11, “Requisite Variety.”

49. Leveson, N., “The Role of the Mental Model,” in *Engineering a Safer World: System Thinking Applied to Safety*, The MIT Press, Cambridge, MA, 2011.

50. de Rosnay, J., “What is a System? Structure and Functional Aspects of Systems,” *Le Macroscope*, Seuil, Paris, 1975.

51. FAA, “Standardized Procedures for Usage of Issue Papers and Development of Equivalent Levels of Safety Memorandums,” Order 8110.112, June 15, 2010.

52. “Verification of Adaptive Systems,” NASA Phase 1 Report, May 2013.

53. Berthoz, A., *Simplexity*: *Simplifying Principles for a Complex World*, Yale University Press, 2012.

54. EASA, “System Design and Analysis,” AMC 25.1309 (Amendment 12), July 2012.

55. AVSI, “User’s Guide, AFE 83 Practical Microcircuit Failure Modeling Tool,” version 1.2, September 25, 2014.

56. Holloway, M.C., “Explicate ’78: Uncovering the Implicit Assurance Case in DO-178C,” 2015.

57. Turchany, G., “La Théorie des systems et systémiques: Vue d’ensemble et definitions,” available at www.prof-turchany.eu/culture/La_theorie_des_systemes.pdf (accessed on 01/11/2016).

58. Unitied States Government Publishing Office. E-CFR Title 49, Subtitle B, Chapter II, Part 236, Subpart H, 236.903. “Definitions,” availalble at http://www.ecfr.gov/cgi-bin/text-idx?SID=87978d327ae6a5916fb53cb4fd296e11&mc=true&node=se49.4.236_1903&rgn=div8.

59. Cornell University Law School. “49 CFR Part 229, Appendix F to Part 229–Recommended Practices for Design and Safety Analysis.” Available at https://www.law.cornell.edu/cfr/text/49/part-229/appendix-F.

60. Cotroneo, D., Natella, R., Pietrantuono, R., and Russo, S., “A Survey of Software Aging and Rejuvenation Studies,” *ACM Journal of Emerging Technologies in Computer Systems*, Vol. 10, No. 1, Article 8, January 2014.

APPENDIX A—COMPARISON OF STANDARDS ON SERVICE HISTORY

A.1 USE OF SOFTWARE SERVICE HISTORY/PRODUCT EXPERIENCE

Table A-1. Comparative description of usage criteria

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
U1	Use of product service history is one of the alternative methods in RTCA DO-178C. An alternative method cannot be considered in isolation from the suite of software development processes. The effort for obtaining certification credit for an alternative method is dependent on the software level and impact of the alternative method on the software life-cycle processes. Guidance for using an alternative method includes showing that it satisfies the objectives of RTCA DO-178C or the applicable supplement; specifying the method in the supporting documentation; and obtaining the agreement from the certification authority.	Use of product service history is one of the additional considerations of design assurance that may be used at the applicant's discretion to satisfy some of the objectives of RTCA DO-254 sections 2–9. Any use of additional considerations should be agreed upon with certification authority. Service experience may be used to substantiate design assurance for previously developed hardware and for COTS components.	The text quantifies that any single alternative method in section 12.3 may be used to satisfy one or more of the objectives of RTCA DO-278A. It also indicates that alternative methods can be combined to support one another. Similar to RTCA DO-178C, alternative methods cannot be considered in isolation from the suite of software development processes, and the level of effort for obtaining approval credit is dependent on the assurance level and impact on the software life cycle.	The acceptability of using one alternative method or a combination of methods is granted by the appropriate approval authority. The justification for using alternative methods is dependent on the safety assessment process.
U2	The acceptability of the method is dependent on the demonstration of an equivalent safety of the software, based on: • Its configuration management. • The effectiveness of problem reporting. • Its stability and maturity. • The relevance of the service history environment. • The length of service history. • The actual error rates in the service history. • The impact of modifications.	The acceptability of the service experience depends on one or more of the following: • Similarity of hardware usage. • Extent to which the design assurance data are based on the proposed configuration. • Extent to which the design errors found during the service period have been addressed. • Actual failure rates in operation.	Criteria number 6 (actual error rates in the service history) for acceptability of the method in RTCA DO-178C, section 12.3.4, is replaced by "the number and severity of failures observed during the product service experience."	–
U3	Sufficiency, relevance, and types of problems occurring during the service history period.	Relevance, usage, and types of problems occurring during the service experience.	Same as RTCA DO-178C.	Only two factors are mentioned: sufficiency and relevance. Problem reporting is not indicated as a criterion.

DO = Delivery Order; EUROCAE = European Organisation for Civil Aviation Equipment; COTS = commercial off-the-shelf

RTCA Delivery Order (DO)-248C DP #4 condenses the necessary conditions to justify the request of certification credit based on software service history to three elements: 1) service period is sufficient, 2) operating environment is the same or similar with additional verification, and 3) product is stable and mature (i.e., few problems are reported/few modifications were performed).

The elements of RTCA DO-254 can be implicitly associated with the more specific elements of RTCA DO-178C. For example, “the extent to which the design errors found during the service period have been addressed” can be associated with both “impact of modifications” and “effectiveness of problem reporting.” The major difference lies in the fact that the list of elements in RTCA DO-178C is inclusive of all items, whereas RTCA DO-254’s list is selective of one or more items. Furthermore, no direct reference is made in RTCA DO-254 to maturity/stability nor the length of service experience; a note in section 11.3 makes a potential correlation between the wide and successful use of an item in service with the notion of confidence in the maturity and stability (inferred from the lack of errors) of the item in service. European Aviation Safety Agency (EASA) Certification Memorandum (CM)-SWCEH-001’s additional guidance elicits that stability and maturity of the component should be supported by evidence, such as number of modifications of the device design or implementation, the nature of the modifications, and the rate of occurrence of errata.

A.2 RELEVANCE OF SOFTWARE SERVICE HISTORY / PRODUCT EXPERIENCE

Table A-2. Comparative description of relevance criteria

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R1	Service history to be used should be defined and agreed upon with the certification authority.	Service experience data relate to data collected from any previous or current usage of the component. RTCA DO-254 explicitly states that service experience data from non-airborne applications are not excluded.	RTCA DO-248C DP#18 elicits some of the tests or evaluation conditions: single or multiple shadow-mode operations and long-duration simulations. In an approach similar to RTCA DO-254, the paper indicates that relevant service experience may be obtained from reuse of software from in-service systems but also from pre-operational systems (e.g., trainers). Finally, the paper clarifies that the word “test” used in RTCA DO-278A is not to be understood as standard testing conditions because it requires additional operational information (e.g., workload).	Sources for relevant service experience data include: reuse of COTS software from in-service ANS systems, or ANS system verification and preoperational activities. The mention of shadow mode and long-duration simulation is associated with sufficiency. Otherwise, the content is similar to RTCA DO-248C DP#18.
R2	The service history data are expressed in a measure relevant to the operations of the system.	–	The text explicitly indicates the type of experience to be accumulated in-service hours or accumulated time under test or evaluation.	Service experience time should be accumulated in-service hours. The mention of event data is an addition only in RTCA DO-278A.

EUROCAE = European Organisation for Civil Aviation Equipment; COTS = commercial off-the-shelf; ANS = Air Navigation Services

Table A-2. Comparative description of relevance criteria (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R3	Software and associated evidence used for compliance demonstration should be shown to have been under configuration management throughout the product service history.	Product service experience data include component identification, its intended function, and DAL.	The document details what RTCA DO-178C refers to as “data” to include not only software but also operating environment and associated evidence used to comply with the safety objectives.	There is no specific mention of demonstration of configuration control for service experience. However, change impact analysis must be done and there is a COTS configuration management process in ED-153 and RTCA DO-278.
R4	Means of collecting and calculating flight hours (software continuously used), or number of demands (on-demand software) should be shown to be sufficiently accurate.	–	The text adds accuracy and completeness of the means for collecting the number of events to account for items not operating continuously. RTCA DO-248C DP#18 adds to the complementary text for RTCA DO-178C in DP#4 that the service experience should cover all operating modes or states (see also R9).	There is no specific mention of accuracy and completeness for qualifying the collection methods. The standard lists processes that can be used for the collection in software with no precedence in ANS application: validation, operator training, qualification testing, operational evaluation, and field demonstration.
R5	Means of collecting and calculating flight hours (software continuously used) or number of demands (on-demand software) should be shown to be sufficiently complete.	–		
R6	Service period should account for changes in any factor important to intended application (e.g., software and system configuration, operational mode or state, and operating environment).	–	RTCA DO-278A explicitly refers to both operating time and event data to account for continuous and discrete operations, respectively. The example factors are slightly different from RTCA DO-178C; system workload is added, and system configuration is removed.	Analysis of change is needed, but there is no mention of accounting for changes with respect to specific factors, as in RTCA DO-278A.

EUROCAE = European Organisation for Civil Aviation Equipment;DAL = Design Assurance Level; COTS = commercial off-the-shelf; ANS = Air Navigation Services

Table A-2. Comparative description of relevance criteria (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R7	Configuration changes should be identified.	–	Same as RTCA DO-178C: software changes need to be identified. Same as RTCA DO-248C DP#4 additional details.	For software previously developed with ED-153, this is included in the process. No specific mention for software with no previous ANS application.
R8	Potential alteration in applicability of history prior to the change should be investigated.	–	Similar to RTCA DO-178C, an analysis is to be performed on whether the changes altered the applicability of the experience; in addition, RTCA DO-278A specifies that uncontrolled changes to executable object code may invalidate the experience data. RTCA DO-278A introduces in this section that change management processes need to be assessed with respect to the appropriate testing level of safety-related problems.	Similar to RTCA DO-178C, any changes made to COTS during the service experience time should be analyzed to determine whether these changes have altered the applicability of the service experience. No further details, as in RTCA DO-278A, are provided.

EUROCAE = European Organisation for Civil Aviation Equipment; ANS = Air Navigation Services; COTS = commercial off-the-shelf

Table A-2. Comparative description of relevance criteria (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R9	Analysis should be performed to ensure that targeted software capabilities are exercised in all operational modes.	The relevance of previous applications should be assessed with respect to the target application, based upon engineering analysis. [For DAL A and B functions only] The analysis should demonstrate that the reused component has been sufficiently exercised.	Same as RTCA DO-178C. Same as RTCA DO-248C DP#4.	Same as RTCA DO-178C.
R10	Analysis should be performed to ensure that relevant permutations of input data are executed.		Similar to RTCA DO-178C, relevant permutations of input data need to be executed. RTCA DO-278A further indicates that if the usage in the service experience data prevented some classes of abnormal inputs from reaching the software, either show that the intended application will preserve the usage in experience data or supplement the experience data with additional testing.	Same as RTCA DO-178C without the additional information in RTCA DO-278A.
R11	Operating environment in service history should be assessed for relevance against the software's intended use.	The relevance of previous environments should be assessed with respect to the target environment, based on engineering analysis.	RTCA DO-278A provides an approach when environments differ in the application of additional verification activity to confirm compliance with the safety objectives. Clarifications same as RTCA DO-248C DP#4.	The text is slightly different. Though the operating environment still needs to be assessed to show relevance to the intended use of the ANS application, the additional verification in case of differences relates to proving the software will operate as intended in the target environment; it does not refer to compliance with safety objectives. But it aligns with RTCA DO-248C DP#18.

EUROCAE = European Organisation for Civil Aviation Equipment; DAL = Design Assurance Level; ANS = Air Navigation Services

Table A-2. Comparative description of relevance criteria (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R12	For credit based on compatibility with hardware environment, the relationship between the service history and target environment should be addressed and the impact of any hardware modification.	The relevance of previous installations should be assessed with respect to the target installation, based on engineering analysis.	Same as RTCA DO-178C. Same as RTCA DO-248C DP#4.	–
R13	Analysis should show that any deactivated code during the history period will not be active in the new environment.	–	RTCA DO-278A has a different approach (more encompassing) to deactivated code. First, it considers the case when the target software is a subset of the software providing the service experience data and points to the demonstration of equivalency of the operating environments. Second, after having identified all deactivated code components during normal operations in the software providing service experience data, the requirement is to show that whether this code is invoked or not in the new environment, its effect is acceptable (i.e., having no adverse effect on system operations). This is less restrictive than in RTCA DO-178C. The added test from RTCA DO-248C is the same in both DP#4 and DP#18.	There is no mention of deactivated code; rather, unused code in the COTS is referenced. Any COTS capability that is not necessary to meet the ANS requirements should be shown to have no adverse effect on ANS operations.

EUROCAE = European Organisation for Civil Aviation Equipment; COTS = commercial off-the-shelf; ANS = Air Navigation Services

Table A-2. Comparative description of relevance criteria (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
R14	–	–	If certification credit is sought for recovery mechanisms implemented in the software or interfaces with the software, the service experience data should support: • The definition of a successful recovery from failure. • The method to calculate the effectiveness of the recovery mechanism and estimation of recovery time. • The identification and analysis of the root causes for unsuccessful recoveries and impact analysis of such failed recoveries on the intended application.	–

EUROCAE = European Organisation for Civil Aviation Equipment

RTCA DO-248C DP#4 specifies the extent of service history and that the appropriate measures of service history should be defined and agreed on with the certification authority. When offering details on sufficiency, the DP further indicates that measurement units should consider both continuous operations and discrete (on-event) operations. Measures could include number of take-offs/landings, flight hours, flight distance, total population operating time, and number of queries.

The type of service experience data are indicated in RTCA DO-254 but not the acceptable units of measurement, though it does cover a large pool of data types; conversely, RTCA DO-178C specifies the units of measurement but not the type of data. EASA CM-SWCEH-001 provides guidance, in addition to RTCA DO-254, for commercial off-the-shelf (COTS) hardware components reused in functions of all assurance levels above D. In this document, the units are explicitly mentioned as operating hours, number of execution hours, and usage duration in years.

RTCA DO-248C DP#4 states that to show R4 and R5, four elements need to be considered:

1. The measurement method applied to compute operating time or number of events.
2. The reliability of that measurement method.
3. The reporting method for operating time or number of events.
4. The reliability of the reporting means.

Note that the DP allocates to this section the consideration of the impact of portions of software not active or not used during the service history, because no certification credit can be requested for these components. The consideration of the methods used to collect service history is only explicit in RTCA DO-178C and RTCA DO-248C—both from a methodology and quality standpoint.

RTCA DO-248C DP#4 provides more details to comply with R7: The identification of changes should be performed on configuration items and documented using records. The types of changes include discrete changes to add functionality or correct errors; any change to the executable object code; or parameter data item files. In addition, evidence should be provided regarding the integrity of the processes used to perform these changes. When judging the adequacy of the change management for all software components for which certification credit is sought using service history, the following criteria should be applied:

- The evidence that all components have been change-controlled throughout the service history is complete and reliable.
- The modifications are described in terms of number and significance.
- The method by which the validity of modified software under consideration is established.

With respect to R8, the DP indicates that, in general, software or hardware changes could alter the applicability of the service history data prior to the change (see also R12). Statements about change management, in the context of product service experience, are not explicit in RTCA DO-254. However, from RTCA DO-178C, R8 can be inferred as any hardware change may potentially impact the applicability of the service experience prior to the change. Because change history needs to be provided as part of product experience assessment data, R7 can be inferred that configuration changes should be identified.

RTCA DO-254 further defines that the substantiation of relevance is derived from engineering analysis and points to the type of documents in which information to substantiate the similarity of usage can be found: in particular, specifications, data sheets, application notes, service bulletins, user correspondence, and errata notices. Note that only RTCA DO-254 is providing a definition for the meaning of similarity (see glossary, main report, section 8).

RTCA DO-248C DP#4 defines the objective of the analysis in R9 and R10 as showing that the software will be performing the same function in the target application as it did in the service history period. The characteristics to be investigated that describe the function are specific to the application. The analysis should be reported in the supporting documentation (see appendix D.1).

The DP clarifies what should be considered in R11 under the criteria of relevance, including:

- Whether or not a difference in usage of the technology in the target environment may change the occurrence of problems (adding or subtracting) or impact the existing system-level recovery mechanisms. The example provided refers to manual versus automatic operations, which can also be seen as an operational mode covered under R9.
- Whether or not changes in operating environment within the service history are documented and analyzed to determine if they render portions of the service history irrelevant.

At equipment level, the comparison of environments is supported by the comparative assessment of the system-level hazards. The objective of the assessment is to indicate whether or not the software assurance level is likely to increase, which may require additional evidence to comply with the safety objectives in the target environment. In line with RTCA DO-254, the comparative assessment needs to include any consideration of change in the installation (see also S2 in table A-3). Because service experience data from non-avionics domains are acceptable, EASA CM-SWCEH-001 explicitly states that the target market for the COTS should be indicated and the specific environment and number of operating hours in which the experience data were gained (e.g., civil or military aircraft; space; telecom; automotive; medical; or consumer).

Finally, the DP addresses the similarity in operating environment in terms of hardware (both under item (3) corresponding to R11 and item (4) corresponding to R12) and points to the elements to be analyzed for similarity, including:

- Resources–Time, processor, memory, accuracy, precision, and communication services.
- Fault prevention and detection mechanisms–Built-in tests, fault-tolerance, and error recovery actions.
- Architecture–Channels and ports; queuing modes; and prioritization.

Any change of hardware or change of software to adapt to the target hardware environment needs to be analyzed and may invalidate the service history data prior to the change.

A.3 SUFFICIENCY OF SOFTWARE SERVICE HISTORY/PRODUCT EXPERIENCE

Table A-3. Comparative description of sufficiency criteria

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
S1	The required amount of service history data is dependent on the system safety objectives of the software and software assurance level.	The product service experience is dependent on the extent to which the design assurance data are based on the proposed configuration of the hardware item.	Similar to RTCA DO-178C, the sufficiency of service experience is conditioned on the system safety objectives and assurance level. RTCA DO-278A introduces an additional qualitative assessment as the level of confidence required that the safety objectives have been met; this dependency is elicited for RTCA DO-178C in RTCA DO-248C DP#4.	Based on engineering judgment and experience with the operation of ANS application, the following approach is proposed to quantify service experience: • Cannot be applied for SWAL1. • 1 year/8760 hours of service minimum with no failure for SWAL2. • Six months/4380 hours of service minimum with no failure for SWAL3. • SWAL4 objectives are typically satisfied without needing alternative methods. The output of reliability models cannot be used to substantiate the sufficiency of service experience time.
S2	The required amount of service history data is conditioned upon the potential differences in the service history environment and system operational environment.	–	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.
S3	The required amount of service history data depends on the objectives being addressed using service history.	–	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	See the proposed quantitative criteria for minimum sufficiency in S1.
S4	The required amount of service history depends on evidence, other than service history data, that addresses a same safety objective.	–	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	The standard acknowledges the possibility that available service experience data may not be able to satisfy all of the objectives, but the proposed quantitative criteria (see S1) are minimum values.

EUROCAE = European Organisation for Civil Aviation Equipment; ANS = Air Navigation Services

RTCA DO-248C DP#4 adds that the required amount of service history will depend on the required confidence interval that the safety objectives have been met (e.g., 95% confidence). The method to compute the amount of service history for the target confidence level is to be documented and accepted by the certification authority. The paper further points to industrial and military reliability standards.

Both standards link the sufficiency to the safety objectives: RTCA DO-178C adds Design Assurance Level (DAL) and RTCA DO-254 points to similarity in configuration. The notion of sufficiency is explicit in RTCA DO-254, Appendix B, covering service experience data for DAL A and B functions. Moreover, in EASA CM-SWCEH-001, experience data for DAL A, B, and C functions can be qualified as sufficient based on the following operating hours (see table A-4) in various environments.[3]

Table A-4. Quantitative criteria for product service experience sufficiency assessment [A-1]

	Sufficient Product Service Experience	Minimum Amount of Usage
DAL/IDAL A	At least two years of use with over 10^6 hours in (aircraft + safety) applications or at least two years of use with over 10^5 hours in (aircraft + safety) applications and over 10^7 hours in other applications.	At least two years of use with over 10^6 hours in (aircraft + safety + other) applications.
DAL/IDAL B	At least two years of use with over 10^5 hours in (aircraft + safety) applications or at least two years of use with over 10^4 hours in (aircraft + safety) applications and over 10^7 hours in other applications.	
DAL/IDAL C	Over 10^5 hours in (aircraft + safety + other) applications.	–

RTCA DO-248C [A-2] frequently asked question #18, “Since there is no specific guidance for handling changes to the aircraft’s operational environment, what part of RTCA DO-178C addresses this type of change?” answers that changes to the operating environment are to be included under the term “aircraft installation” in the requirements in RTCA DO-178C, section 12.1.2. RTCA DO-248C DP#4 indicates that if differences are found in the operating environment, additional service history/other techniques may be needed. RTCA DO-254 primarily ties the operating environment with the assessment of relevance in usage. The notion of length of service experience is not explicit in the standard.

RTCA DO-248C DP#4 recalls that using other techniques in conjunction with service history may reduce the amount of service history needed to meet the safety objectives—quantifying that reduction, however, depends on the type and completeness of the other techniques used. RTCA

[3] Aircraft applications include aircraft operations in flight/on the ground and board/line replaceable unit/system/aircraft tests. Safety applications include space, airborne military, nuclear, medical, railway, and automotive. Other applications include banking, computer, and telecommunications.

DO-254 mentions additional evidence, such as any available statistics related to the design errors, but does not relate them to a reduction in the recommended length of service experience. In the context of service experience, these statistics support the problem-reporting considerations. Additional verification strategies are mentioned for components used in DAL A and B functions but not with the objective of reducing service experience.

A.4 CONSIDERATIONS OF PROBLEM REPORTING

Table A-5. Comparative description of items related to problem reporting

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
P1	The list of specific data should be agreed upon with the certification authority (include items in sections 11.17 and 12.2.4.3).	The relationship between problem reports and hardware item or product requirement changes should be established (or assessed).	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	–
P2	The chronological trends of problem reports should be evaluated and any increasing trend explained.	Problem reports may show that service experience has led to improvement now available in the current configuration.	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	–
P3	The completeness of the software error history should be assessed based on logs, collection/reporting means, statistical data, etc.	The completeness of the problem report coverage may be assessed through the established relationships between problem reports and hardware item or product requirement changes.	Same as RTCA DO-178C and RTCA DO-248C DP#4/DP#18.	–
P4	Problems indicative of inadequate process (e.g., design errors, code errors) should be indicated separately from problems outside the scope of RTCA DO-178C.	The acceptability of service experience depends on the extent to which design errors found during the service period have been eliminated, mitigated, or analyzed and determined to have no safety impact on the configuration to be used.	RTCA DO-278A clarifies that process-related problems need to be sorted between design/code errors and problems having their cause outside the scope, such as hardware failure and system requirements errors. The explanations in RTCA DO-248C are the same for both RTCA DO-178C (DP#4) and RTCA DO-278A (DP#18).	–

EUROCAE = European Organisation for Civil Aviation Equipment

Table A-5. Comparative description of items related to problem reporting (continued)

Ref	RTCA DO-178C	RTCA DO-254	RTCA DO-278A	EUROCAE/ED-153
P5	Safety-related problems should be identified and an evaluation performed to confirm they have all been corrected.	Any available statistics on design errors and their impact on the safety assessment process should be evaluated. Problems identified but not fixed may still be mitigated by architectural means or performing additional verification.	RTCA DO-278A has a broader and more detailed approach to safety-related problems. It requires the evaluation of these problems for potential adverse effects on intended operations and the recording of any problem implicating software for which the effect is not consistent with the safety assessment as they will be considered to be failures. Finally, failures should be considered for potentially invalidating the service experience data for any time preceding the correction of the associated problem. The explanations in RTCA DO-248C are the same for both RTCA DO-178C (DP#4) and RTCA DO-278A (DP#18).	Same as RTCA DO-278A.

EUROCAE = European Organisation for Civil Aviation Equipment

RTCA DO-248C DP#4 points to six aspects of the problem-reporting process to be considered for assessing whether the data support the substantiation of P1–P3:

1. The method used for detecting in-service problem and its reliability.
2. The method used for recording in-service problem and its reliability.
3. The method for determining the severity of the in-service problem.
4. The method used for differentiating in-service problems attributable to software from other causes.
5. If applicable, the method for collecting non-service problems and assessing their relevance.
6. The method for assessing the impact of product improvements or unresolved problem reports (if any) on the credit claimed using service history.

The DP highlights two types of problems to which special considerations should be given as they impact the way reliability data are built:

1. Common cause problems–Collocated in time, potential high-impact in terms of severity.
2. Cascading problems–Clustered in time, potential higher impact because of their aggregation, not their number of individual constituents.

Similar to issues related to the methods of collection of service history, RTCA DO-254 does not place any consideration on the methodology associated with problem reporting—whether it pertains to collecting the reports or assessing them.

Problems that are out of scope of RTCA DO-178C include system design errors, hardware design errors, and hardware failures. Hardware design errors are within the scope of RTCA DO-254. RTCA DO-248C DP#4 discusses how to address software process inadequacies unearthed through the analysis of service problem history. The process that introduced or failed to detect the errors can be suspected to be inadequate and need to be further analyzed before the decision to proceed, or to invalidate the service history, can be made. The assessment should consider characteristics such as:

- A large proportion of defects may be indicative of a software development process issue.
- Failure related to off-nominal inputs may indicate issues with software robustness either in the requirement development, or in the verification, testing.
- Presence of partially fixed problems may point to the design, development of requirements, or regression testing.
- Errors being reintroduced may result from issues with configuration management.
- Failures caused by borderline values may have resulted from design issues (e.g., improper interface) or coding (e.g., numerical processing problem).

The service history may remain valid if the inadequacy in the process can be shown to be:

- An isolated event–An adequate process was not followed in only one instance (the applicant will show that the process is capable of detecting errors of the same type).
- An identifiable deficiency or omission–Additional analysis, inspection, or other verification activity will be needed and an analysis to determine whether there are other latent errors related to the process inadequacy.

In addition to the information in the DP, RTCA DO-178C is more detailed regarding the types of process-related issues to be investigated and the conditions under which the service history may remain valid. No such details are provided for hardware under RTCA DO-254.

RTCA DO-248C DP#4 provides additional guidance pertaining to the actions to be performed when safety-related problems are identified and traced to an inadequate software development process:

- The gaps in the inadequate process need to be identified.
- The product needs to be further analyzed to determine whether other safety-related problems exist.
- Additional verification activity needs to be performed to satisfy the required level of assurance associated with the safety objectives to which the missing or incomplete elements referred in the inadequate process.

The DP addresses in more details the confirmation of correction:

- The service history may not be used for credit if the inadequate process is the software development process and the correction of the safety-related problems, and reverification process could either introduce new defects or alter the results of the quantitative analysis.
- Conversely, the service history may remain valid after resolution of a safety-related problem if it can be shown that the problem was not the result of a systemic cause.

With the complement of information in the DP, RTCA DO-178C provides more details on the actions to be performed to address a residual safety-related problem. Both documents indicate the need for additional verification for issues not corrected. RTCA DO-254 indicates other means to address residual safety-related problems via mitigation at the architecture level, whereas RTCA DO-178C focuses on situations that may invalidate the service history.

A.5 SUPPORTING DOCUMENTATION

The evidence supporting the assurance case is documented in a Plan for Software Aspects of Certification (PSAC) for software and in a Plan for Hardware Aspects of Certification (PHAC) for hardware. The following are explicit requirements regarding the content of these documents relative to service history/experience assessment data (see table A-6).

Table A-6. Comparative description of documentation items

Ref	RTCA DO-178C - PSAC	RTCA DO-254 - PHAC	RTCA DO-278A	EUROCAE/ED-153
D1	Rationale for relevance: • All items in section 2.2 of this document or section 12.3.4.1 of RTCA DO-178C.	Rationale for relevance: • Similarity of hardware item usage with respect to application, function, operating environment, and DAL. • Detailed service information being considered. • Change history.	All items for relevance, same as RTCA DO-178C.	–
D2	Amount of service history: • All items in section 12.3.4.2 of RTCA DO-178C. • Any censoring rules applied to the service history data. • Measured parameters (in units relevant to the operations).	Description of the service experience data collection and assessment process, including criteria for determining the adequacy and validity of the data. Justification for the sufficiency of service experience data (for DAL A and B functions only).	All items justifying sufficiency, censoring rules, and measured parameters—same as RTCA DO-178C.	–
D3	Rationale for calculating total relevant service history: • Operational modes. • Number of independently operating copies. • Definition of normal operation and normal operation time. • Analysis of error rates if greater in service data than in the plan.	Extent to which the design assurance data are based on the proposed configuration of the hardware item. Justification of the adequacy of the service experience data relative to the intended use and required assurance level. Assumptions used to analyze the service experience data.	RTCA DO-178C refers to this item as "rationale for calculating total relevant service history," while RTCA DO-278A refers only to "rationale for calculating the number of hours in service," relevance is implicit, and total is not indicated. In addition to the elements common with RTCA DO-178C, RTCA DO-278A clarifies the rules for calculating service experience time when using several operating copies, which is not covered in such detail in RTCA DO-248C DP#4.	Similar to RTCA DO-278A, the standard indicates that the accumulated in-service hours need to take into account the number of copies in service when the associated environment is relevant. Associating each copy to a pre-negotiated percentage of the total in-service hours is the rule listed in RTCA DO-248C DP#4.

EUROCAE = European Organisation for Civil Aviation Equipment;

Table A-6. Comparative description of documentation items (continued)

Ref	RTCA DO-178C - PSAC	RTCA DO-254 - PHAC	RTCA DO-278A	EUROCAE/ED-153
D4	Definition of what counted as an error and rationale for the definition.	–	Same as RTCA DO-178C.	–
D5	Proposed acceptable error rates and rationale for history period.	Actual failure rates in operation.	In line with the modification in U2 for the use of service experience data, RTCA DO-278A replaces “proposed acceptable error rates” with acceptable maximum number and severity of failures—with a justification for both the service experience period and the number of failures vis-à-vis the safety objectives.	–
D6	Definition of criteria for problems that would invalidate service history.	–	Same as RTCA DO-178C.	–
D7	Criteria for errors that will be corrected; how they will be corrected and verified; and rationale for defects for which no action will be taken.	Extent to which the design errors found during service period being assessed have been eliminated, mitigated, or analyzed and determined to have no safety impact on the configuration to be used.	Same as RTCA DO-178C.	–
D8	Objectives in sections 4–9 to be addressed through the use of service history.	–	Same as RTCA DO-178C.	–

EUROCAE = European Organisation for Civil Aviation Equipment

In addition, EASA CM-SWCEH-001 [A-1], section 8.3, explicitly states that the service experience for application-specific integrated circuit and programmable logic device electronic hardware should be documented in the PHAC.

RTCA DO-248C DP#4 details the elements to be analyzed for discrepancies when justifying the computation of the total relevant service history (D3):

- Total operating time and workload by operating mode (if applicable).
- Workload by function per operating mode.
- Proportion of failures in the software functionality that will not be used in the target application.

The DP comments on the rules for calculating service experience time when using several operating copies found in RTCA DO-278A, as ensuring that at least one copy is operated long enough to substantiate stability claims. RTCA DO-254, Appendix B section 3.2.3, addresses D8 but is limited to hardware components reused in DAL A and B functions—first, in the negative, by recommending the documentation identify where the service experience data are not sufficient to complete the design assurance. Then, in the positive, by recommending an explicit traceability is shown between the service experience data and the verification data supporting the demonstration of coverage of design assurance.

Other safety-related domains address service history using a set of criteria that are similar enough to the ones for the aviation domain that they can be reused. However, the information provided in the standard does not address all of the criteria defined in section 2.1 as use (U), relevance (R), sufficiency (S), problem reporting (P), and supporting documentation (D).

Table A-7. Comparison of use criteria for other safety domains

Ref	Nuclear Industry	Medical Devices	Rail
U1	Experience databases can be used to substantiate development assurance for COTS. COTS software is expected to have been developed on good software engineering practices.	The use of OTS software in a medical device depends on the results of the hazard analysis. The extent of analysis and documentation (e.g., basic or special) is dependent on the level of concern.	Europe/U.K.–Comparison with similar systems is one of the three risk acceptance principles of the CSM (the others being application of codes of practice and explicit risk estimation). Though any of the three risk assessment principles can be used to perform the hazard analysis, it is likely that a combination of all will be used with most major projects. U.S.–Other methods, including in-service experience, may be acceptable if demonstrated to be equally suitable to the satisfaction of the Associate Administrator for Safety. Previous approval or recognition of a train control system, together with an established service history, at the request of the positive train control railroad and consistent with available safety data, may be credited toward satisfaction of the safety case requirements set forth in 49 CFR 236 subpart I with respect to all functionalities and implementations contemplated by the approval or recognition.

OTS = off-the-shelf; CSM = common safety methods; CFR = Code of Federal Regulations;

Table A-7. Comparison of use criteria for other safety domains (continued)

Ref	Nuclear Industry	Medical Devices	Rail
U2	The acceptability of experience databases is dependent on the level of criticality of the system, its configuration management, and the relevance of the environment and usage.	The basic acceptability of OTS software is dependent on version control, existence of OTS manufacturer documentation, and the relevance of usage.	Europe/U.K.–For commonly accepted codes of practice, the CSM recognizes not only railway sector standards but also codes of practice from other domains, including aircraft if proof can be made of their acceptability. Deviations are acceptable if the applicant (called the proposer) can demonstrate that at least the same level of safety can be achieved using an explicit risk estimation, other codes of practice, or similar systems (also called reference systems). In-service history is associated with reference systems. For an existing system to be usable as a reference to derive safety requirements for new or change systems, the following conditions need to be demonstrated: • The existing system has been proven in use and has an acceptable safety level. • It is accepted in the member state when the change is to be introduced. • The target system is used under similar functional, operational, and environmental conditions and has similar interfaces as the reference system. The following caveat is indicated in the UK guidance: it is unlikely that evidence of in-service history alone can satisfy the demonstration of acceptable safety level for technical changes proposed to a high-integrity system based on the low failure rates required. U.S.–Service history plays an indirect role in the evaluation criteria for the performed risk assessment because it depends on: • The extent to which recognized standards have been used in product design and the relevant safety analysis. • The availability of quantitative data (including statistical confidence levels using accepted methods) associated with risk estimates. • The complexity of the product and extent to which it will incorporate or deviate from design practices associated with previously established histories of safe operations. • The degree of rigor and precision associated with the safety analyses (e.g., comprehensiveness and sensitivity analysis). • The extent to which validation of the product has included experiments and tests to identify uncovered faults in the operation of the product. • The extent to which identified faults are effectively addressed. • Whether the risk assessment for the previous condition was conducted using the same methodology as that for operation under the proposed condition. • If an independent third-party assessment is required or performed at the election of the supplier or railroad, the extent to which the results of the assessment are favorable. Field testing of the product may be conducted prior to the approval of the safety plan.
U3	Not covered.	Relevance of usage is primarily mentioned. Problem reporting is explicitly mentioned for major level of concern.	Not covered.

OTS = off-the-shelf; CSM = common safety method

Table A-8. Comparison of relevance criteria for other safety domains

Ref	Nuclear Industry	Medical Devices	Rail
R1	The type of data deemed relevant is not detailed. However, it is recommended that the experience database be statistically valid. This is different from the recommendation that all modes be exercised (R9) and all input permutations executed (R10).	Not covered	Not covered
R2	Not covered	Not covered	Europe/U.K.–Risk levels must be expressed in units of consequences per unit of exposure for the previous conditions and for the life cycle of the product. Exposure is total train miles traveled per year over the relevant infrastructure. Units of consequences identify total cost (including fatalities, injuries, property damage, and other incidental costs) resulting from preventable accidents associated with the functions performed by the system. For passenger traffic, second risk metrics use passenger miles per year and total societal costs of passenger injuries and fatalities.
R3	The experience database needs to be checked against the COTS software version; this activity requires a tracking of the COTS software version.	Design records integrate the version of the relevant OTS software. When changes are made, the record needs to be updated. Configuration management is performed within the computer system specifications for hardware, software, and problem correction (e.g., patches).	U.S.–Since June 2005, software for signal and train control systems must be under configuration management throughout the life cycle of the system. Once in operation, the hardware, software, and firmware are version controlled in the configuration management control plan and operations and maintenance manual. The date is March 2010 for locomotive electronics under 49 CFR 229.
R4	Not covered	Not covered	Not covered
R5	Not covered	Not covered	Not covered
R6	Not covered	Not covered	Not covered

OTS = off-the-shelf; CFR = Code of Federal Regulations

Table A-8. Comparison of relevance criteria for other safety domains (continued)

Ref	Nuclear Industry	Medical Devices	Rail
R7	See R3	See R3	U.S.–The configuration changes are identified and tracked using track records within the operating manual.
R8	Not covered	Not covered	Europe/U.K.–Technical changes need to be analyzed for scope and safety-related elements. The safety-related elements will be assessed for significance. The criteria to determine significance include the consequence of the failure (worst case scenario), level of innovation (associated with uncertainty in outcome), complexity of the change, ability to monitor the implemented change throughout the life cycle, reversibility of the change, and assessment of the change significance in view of all recent safety-related but less significant modifications. A significant change needs to be analyzed using the CSM, but no threshold is provided to make the decision on significance. U.S.–If a processor-based signal and train control system is to be reused in an environment where either the physical or operating conditions might change prior to the implementation or during the life cycle, there should be adjustments. These adjustments may be additional safety barriers of several forms (e.g., additional control system or signal in the cab). The adjustments are predicated on the traffic volume increase and operating speeds.

CSM = common safety method

Table A-8. Comparison of relevance criteria for other safety domains (continued)

Ref	Nuclear Industry	Medical Devices	Rail
R9	Not covered.	The similarity in usage and environment needs to be analyzed, particularly when design changes are introduced. The analysis can be based on documentation but also on the performance of verification and validation. Elements to be analyzed include resource requirements, timing, memory organization, built tools, data integrity issues, and human factors (see for comparison RTCA DO-248C DP#4). For certain applications of medical devices, installation is also a factor to consider in assessing the similarity of the previous and target environment.	Not covered.
R10	Not covered.	Not covered.	U.S.–Range input testing is required to perform the sensitivity analysis that will support the achievement of a "high level of confidence."
R11	The operating environment and usage associated with the experience database are exactly or nearly the same as for the target environment.	Not covered.	U.S.–Consideration of similarity for environment is mentioned relative to two cases: • In case of hardware change, the demonstration of similar characteristics must be done in the historical environment or target environment. • If the environment has significantly changed when a positive train control system is to be changed, the analysis shall proceed as would be the case prior to approval or recognition of the system.
R12	Not covered.	Not covered.	Not covered.
R13	Not covered.	The guidance document does not refer to deactivated or unused code, but rather to non-specified OTS software, unwanted system use, or adding unwanted software. Analysis should show that measures are in place to prevent the operation of any non-specified OTS software. Solutions include system design, preventing measures, labeling, or disabling input (e.g., compact disk).	Not covered.

OTS = off-the-shelf

Table A-9. Comparison of sufficiency criteria for other safety domains

Ref	Nuclear Industry	Medical Devices	Rail
S1	The statistical validity of the experience database is a function of the software application criticality. Service history is not expected to compensate for poor or lack of software engineering best practices. At the highest level of criticality, the U.S. Nuclear Regulatory Commission does not accept software service history.	There is no guidance regarding sufficiency. More extensive data experience is recommended as a function of level of concern (e.g., major).	U.S.–Service history use may support both full risk and abbreviated risk assessments. The requirements for full risk assessment are more stringent than for abbreviated risk assessment. The full risk assessment must address the safety risks affected by the introduction, modification, replacement, or enhancement of a product, including risks associated with the previous condition that are no longer present as a result of the change, new risk not present in the previous conditions, and risk neither newly created nor eliminated but affected by the change. The abbreviated risk assessment can be done if no new hazards are introduced by the change, the severity of each hazard for the previous condition does not increase, and exposure to such hazards does not change from the previous condition.
S2	Not covered.		Not covered.
S3	Not covered.		U.S.–Service history of positive train control systems can be used to show that the freight or passenger operation (above 125 miles per hour) will be operated at the level of safety comparable to that achieved over the 5-year period prior to the submission of the safety plan by other train control systems that perform positive train control functions (see 49 CFR part 236 subpart I) and which have been used on high-speed rail systems with similar technical and operational characteristics in the U.S. or foreign service, provided that the use of foreign service data must be approved by the associate administrator before submittal of the safety plan.
S4	Not covered.		Europe/U.K.–For high-integrity systems, service history shall be complemented with other CSMs (i.e., accepted codes of practice/explicit risk estimation). U.S.–Previously approved positive train control systems can apply for an expedited certification if they have been in operation for at least 5 years and assessed by an independent third party (or have received a waiver).

CSM = common safety method; Code of Federal Regulations

Table A-10. Comparison of problem-reporting considerations for other safety domains

Ref	Nuclear Industry	Medical Devices	Rail
P1	The statistical validity of the experience database is a function of the software application criticality. Service history is not expected to compensate for poor or lack of software engineering best practices. At the highest level of criticality, the U.S. Nuclear Regulatory Commission does not accept software service history.	The list of specific data is indicated as part of the recommendations for producing special documentation for applications with a major level of concern.	Europe/U.K.–The considerations of problem reporting are associated with the proposer's duty to create and maintain a track record (no particular format is mandated in Europe) that will be used if the system is used as a reference system in the future. U.S.–Problem-reporting considerations are included in the safety plan within post-implementation records for both routine maintenance and testing and component failures resulting in safety-relevant hazards. Any inconsistency between the frequency of a safety-relevant hazard and the value in the safety plan must be reported, and countermeasures must be taken. The safety plan contains a list of predefined changes, which are not considered design modifications, and need only be shown to satisfy the minimum performance standard. The safety plan must identify control measures designed to ensure that predefined changes will not compromise safety-related functional requirements and safety-critical hazard mitigation processes.
P2	Not covered.	Not covered.	Not covered.
P3	Not covered.	The impact of the introduction of new or modification of existing OTS software in a medical device's baseline needs to be assessed using failure mode and effect analysis. In addition, experience data should be included in the form of release bulletins (reporting known errors), user manuals, specifications, patches, literature, and Internet searches for other user's experience with the OTS software.	Not covered.

OTS = off-the-shelf

Table A-10. Comparison of problem-reporting considerations for other safety domains (continued)

Ref	Nuclear Industry	Medical Devices	Rail
P4	Problem reporting should address all types of failures considered in the safety evaluation, including software and hardware failures, design errors, erroneous inputs, human errors, and mischievous attacks. The standards mainly cover the development process and lack in specific guidance for the measurement of these problems at the local level. The independent evaluator has the responsibility to evaluate the problem tracking on criteria not in the standards.	Not covered.	U.S.–Specific types of problems (failures) are highlighted in the U.S. regulations; in particular, unsafe systematic failures are in scope. For locomotive electronics, the problem reporting must support demonstrating that the product is designed to mitigate or eliminate unsafe systematic failures. Unsafe systematic failures are defined as conditions that can be attributed to human error that could occur at various stages throughout product development (e.g., software errors due to human error in the specification; design or coding; hardware design; human-machine interface; installation and maintenance errors; and errors in making modifications). Particular areas of concern for systems in parts 229 and 236 include: • Systematic failures. • Random failures, including latent failures or non-self-revealing failures leading with a subsequent failure to an unacceptable or undesirable condition. Common mode failures (software, hardware, or both).
P5	Not covered.	For OTS software involved in a major level of concern, data experience can be used to show that workarounds have been developed to address the relevant problem. They do not need to be all corrected—for example, the problems can be mitigated to a lesser level of concern. The guidance provides examples of both hardware and software mitigations.	U.S.–Software faults should not cause unacceptable or undesirable hazards.

OTS = off-the-shelf

A.6 REFERENCES

A-1 “Certification Memorandum on Development Assurance of Airborne Electronic Hardware,” EASA Report CM-SWCEH-001, March 2012.

A-2 “Supporting Information for RTCA DO-178C and RTCA/DO-278A,” RTCA DO-248C, December 2011.

APPENDIX B—STATISTICS OF LIFETIME DATA

The general field of statistical analysis of lifetime data is also sometimes referred to as statistical analysis of survival time or failure time data. The applications are varied from social sciences (using the term lifetime data), to biomedical (preferring the term survival time), and to engineering, in which failure time is the focus of the analysis. Across all fields of application, the definition of lifetime includes a time origin, time scale, and a specification of the event that characterizes the lifetime (e.g., death, failure, social event such as marriage). However, the time scale might not always be chronological time (e.g., number of printed pages for a photocopier).

The objective of modeling the lifetime data and of performing a statistical analysis on them is to achieve an estimation of their distribution so that comparisons can be made, scientific understanding can be furthered, processes can be improved, predictions can be established, and decisions can be made. The models typically use explanatory variables that can be traced to features of the lifetime data [B-1]. The following sections recall basic statistical definitions and define the primary functions used in describing lifetime data; notations and a table of equations provide supplemental information useful in reading the mathematical elements of this report.

Table B-1 summarizes the various notations used within the document. In an effort to provide consistency, some notations had to be changed from their original form in the referenced source document.

Table B-1. Synopsis of mathematical notations

Notation	Description
T	Continuous-time or discrete-time random variable with values t. In context, T models the time to failure with values t_i as time of the i^{th} failure, response variable in a regression model with values t_i.
$F(t)$	Distribution function of the continuous-time or discrete-time random variable T.
$P_r(\ )$	Probability that ( ) in a continuous time sense. In context $P_r(B)$ is the probability that event B occurs.
$m(\ )$	In context, discrete mass function, continuous mean function.
$f(t)$	Probability density function associated with the continuous-time distribution $F(t)$.
E	Moment operator on a random variable. In context: m_n or m_n denote the n^{th} moment. In context, m represents the first moment or the mean/average value.
$var(T)$	Second moment or variance of a random variable T.
$\sigma(T)$	Standard deviation of a random variable T.
$P(A\|B)$	Conditional probability: probability of event A given that event B has occurred.
$\cap$	Intersection (AND) operator on probabilistic events.
$S(t)$	In context, survivor function, reliability function (also indicated as $R(t)$).

Table B-1. Synopsis of mathematical notations (continued)

Notation	Description
$h(t)$	Hazard function. In context of reliability $\lambda(t)$: failure intensity of repairable systems or failure rate of non-repairable systems.
Δt	Interval of time [typically small].
θ	Generic model parameter. $f_\theta(t)$: probability density function of T parameterized by θ. $\hat{\theta}$: the estimated value of θ. θ_i: value of θ obtained from simulation number i.
$L_t(\theta)$	Likelihood function of the model parameter θ given the random value t. $\Lambda_t(\theta)$: logarithm of the likelihood function.
$\frac{\partial}{\partial\theta}$	Partial derivative with respect to θ operator.
n	In context, total number of failures, number of moment.
X	In context, covariate in a regression model with values x_i, sojourn time in a state of a Markov chain.
η	In context, the noise in a regression model.
β	In context, random variable. $\hat{\beta}$: estimated value of the variable β or estimator on β. $bias(\hat{\beta})$: bias of the estimator on β.
$Eff(.)$	Efficiency of an estimator (.). $Eff_R(\hat{\beta}_1, \hat{\beta}_2)$: relative efficiency of the estimator β_1 versus the estimator β_2.
$I(\beta)$	Fisher information function of β.
$\|x\|$	Absolute value of x.
$MTBF_c$	Cumulated mean time between failures (or MTBF over the past).
$MTBF_i$	Instantaneous mean time between failures (or MTBF at present).
$MTBF_f$	Future mean time between failures (or MTBF in the future).
c, ϕ, α, γ, N, a, b	In context, parameters in statistical models.
S	In context, set of jump times with values s_i in a Markov chain.
J	In context, set of states with values J_i in a Markov chain.
Q	In context, transition rate matrix for a Markov chain with values q_{ij}.
P	In context, process transition matrix for a Markov chain with values p_{ij}.
U	In context, state space of “Up” (working) states in a Markov chain.
D	In context, state space of “Down” (failed) states in a Markov chain.
k	In context, number of groups for grouped data, model parameter.
$k(t)$	In context, function used to parameterize the Power Law Process models.

MTBF = mean time between failures

B.1 STATISTICAL PROPERTIES

The following statistical properties are defined in this section:

- Random variable and distribution function
- Moments of a random variable
- Conditional probability

The properties of a random variable *T* with values denoted as *t* are completely described by its distribution function *F(t)* defined as:

$$F(t) \overset{\Delta}{=} P_r(T \le t) \tag{B-1}$$

The distribution function is monotone and non-decreasing with $\lim_{t\to-\infty} F(t) = 0$ and $\lim_{t\to+\infty} F(t) = 1$.

A random variable *T* is called discrete if there exists a mass function *m*$m_t(.)$ such that:

$$F(t) = \sum_{\substack{x \le t; \\ m_t(x) > 0}} m_t(x)\,. \tag{B-2}$$

That is, a discrete random variable can assume only a countable number of values.

A random variable *T* is called continuous if there exists a probability density function *f(t)* such that:

$$F(t) = \int_{-\infty}^{t} f(x)dx\,,\ \text{for}\ -\infty < t < +\infty\,. \tag{B-3}$$

The probability density function exists if the distribution function is absolutely continuous.

Moments are used to specify the random variable. Some moments tie directly with observed characteristics. This section recalls the definition for the moments used in analysis and modeling.

The n^{th} moment of a continuous random variable *T* is defined as:

$$E(T^n) \overset{\Delta}{=} \int_{-\infty}^{\infty} t^n f(t)dt\,. \tag{B-4}$$

Two moments are of particular interest:

- The first moment of *T* is known as the expectation, mean, or average.
- The second moment of *T* is called the mean square value.

The n^{th} moment about the mean or n^{th} central moment of a continuous random variable *T* is defined as:

$$E\left[(T-E(T))^n\right] \overset{\Delta}{=} \int_{-\infty}^{\infty} (t-E(t))^n f(t)dt \,. \qquad \text{(B-5)}$$

The second moment about the mean is of particular interest and is called the variance of *T* or *var(T).* It is a measure of the dispersion about the mean in the samples of *T*. Another description often used is the standard deviation σ, which is defined as:

$$\sigma(T) \overset{\Delta}{=} \sqrt{\text{var}(T)} \,. \qquad \text{(B-6)}$$

Conditional probabilities are used in the description of lifetime data. For example, in modeling the probability that a system will fail given that no failure has been observed up to time *t* or to model the probability that an individual will die given that he's lived *x* years. Given two events, *A* and *B,* the conditional probability function

$$P_r(A\,|\,B) \overset{\Delta}{=} \frac{P_r(A \cap B)}{P_r(B)} \qquad \text{(B-7)}$$

is described as the probability of event *A*, having already observed the occurrence of event *B*. Note that all characteristics described above can be extended to conditional ones.

B.2 LIFETIME DATA DISTRIBUTIONS

Lifetime distributions are represented by continuous models or discrete models. Continuous models are well-suited to social sciences when considering, for example, the lifetimes of human beings.

For continuous-time models, considering a single, continuous, non-negative, random variable *T* defined over [0, ∞) with probability density function *f(t),* define:

$$S(t) = P_r(T \geq t) = \int_{t}^{\infty} f(x)dx \qquad \text{(B-8)}$$

S(t) is interpreted based on the meaning of the characterizing event for the data. If *T* represents the lifetimes of individuals and the characterizing event is death, then *S(t)* is called the survivor function and expresses the probability of an individual surviving to time *t*. If *T* represents the lifetime of a manufactured product and the characterizing event is a failure, then *S(t)* is called the reliability function (often times denoted as *R(t)*) and expresses the probability of a product to experience no failure before time *t*. *S(t)* is a monotone decreasing continuous function with $S(0)=1$ and $S(\infty)=\lim_{t\to\infty} S(t)=0$.

Another important concept applicable to lifetime distributions is the hazard function *h(t)*, defined using conditional probability as:

$$h(t)=\lim_{\Delta t\to 0}\frac{P_r(t\le T<t+\Delta t\,|\,T\ge t)}{\Delta t}=\frac{f(t)}{S(t)} \qquad \text{(B-9)}$$

The hazard function represents the instantaneous rate of the characterizing event: rate of death in a survivor function context or the rate of failure in a reliability function context at a time *t*. Furthermore, *h(t)Δt* represents the approximate probability of the characterizing event in the interval $[t,t+\Delta t)]$ given that it has not occurred up to *t*.

The distribution of *T* can be expressed equivalently by *f(t)*, *F(t)*, *S(t)*, or *h(t)* [B-1] using:

$$S(t)=\exp\left(-\int_0^t h(x)dx\right) \text{ and } f(t)=h(t)\exp\left(-\int_0^t h(x)dx\right) \qquad \text{(B-10)}$$

When lifetimes are grouped or measured via cycles of some sort, *T* may be regarded as a discrete random variable. In this case, the probability function is expressed as:

$$f(t_j)=P_r(T=t_j) \text{ for } j\text{=1, 2,…} \qquad \text{(B-11)}$$

The survivor function is given by:

$$S(t)=P_r(T\ge t)=\sum_{j:t_j\ge t} f(t_j) \qquad \text{(B-12)}$$

S(t) is a left-continuous, non-increasing step function with $S(0)=1$ and $S(\infty)=0$. The discrete-time hazard function is defined as:

$$h(t_j)=P_r(T=t_j\,|\,T\ge t_j)=\frac{f(t_j)}{S(t_j)} \text{ for j=1, 2,…} \qquad \text{(B-13)}$$

As for the continuous time case, *f(t$_j$)*, *S(t$_j$)*, and *h(t$_j$)* can be equivalently used as specifications of the distribution of *T*.

B.3 REFERENCES

B-1 Lawless, J.F., *Statistical Models and Methods for Lifetime Data*, Wiley, 2003.

APPENDIX C—RELIABILITY

This appendix provides background information for section 5.1.2.

C.1 MEAN TIME TO FAILURE AND MEAN TIME BETWEEN FAILURES

C.1.1 DEFINITION OF MEAN TIME BETWEEN FAILURES

For the purpose of mean time between failures (MTBF), the term of failure applies to any condition wherein the system no longer provides its intended functionalities. MTBF is defined as the sum of two quantities: the mean up time and mean down time, as shown in figure C-1.

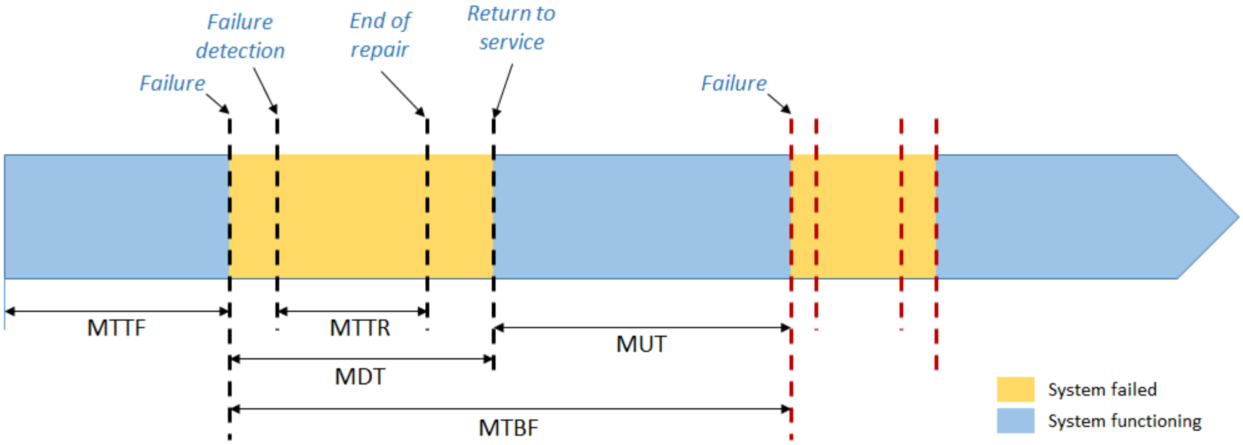

Figure C-1. Components of MTBF

The time between the occurrence of the failure and the detection of the failure may be difficult to estimate in some cases, so MTBF may carry an assumption of immediate detection, or it can include the time between the unscheduled removal after the occurrence of the failure and the diagnosis in the repair center. After the failure is repaired, verification tests are run, and the equipment is returned to operation. The time from insertion to first failure is mean time to failure (MTTF), whereas afterwards for (repairable systems), it is MTBF.

C.1.2 VARIATION OF MTBF OVER TIME

MTBF vary over time. With respect to the bathtub curve in figure 16 from the main report, MTBF can be:

- Constant in time, meaning that the system is mature.
- Decreasing in time, meaning that the system deteriorates.
- Increasing in time, meaning that the system improves.

Three types of MTBF have been defined as the following: 1) cumulated MTBF, 2) instantaneous MTBF, and 3) future MTBF.

Figure C-2 shows the cumulated MTBF for an observation time at T_k. This value takes into account the past values of MTBF computed at T_{k-3}, T_{k-2}, and T_{k-1}, so that the observation period covers from entry into service to T_k.

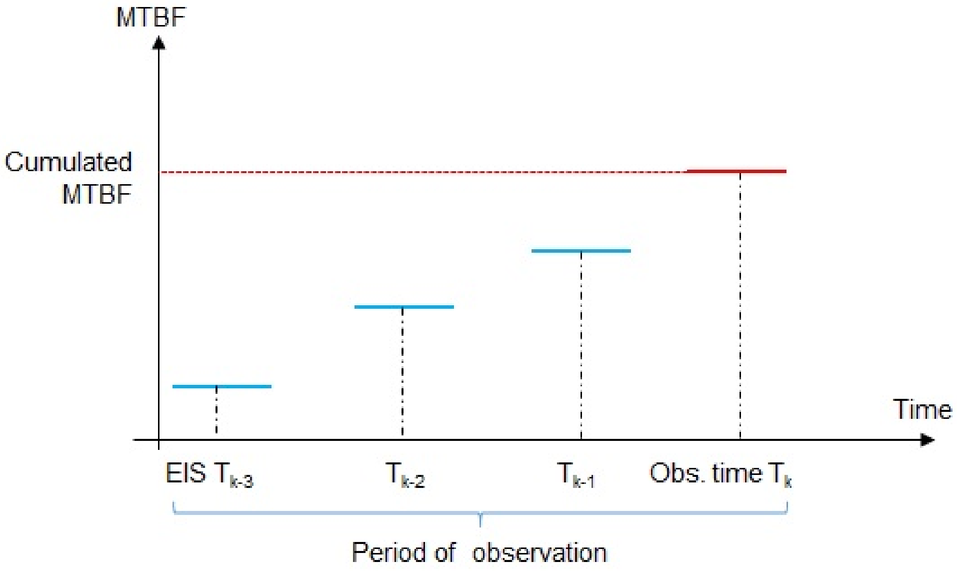

Figure C-2. Cumulated MTBF

Figure C-3 shows the future MTBF as the forecasted mean time to the next failure. This value is predicted from past values of MTBF at T_k-2 and T_k-1, and the present value computed at T_k, so that the observation period includes up to T_k+1.

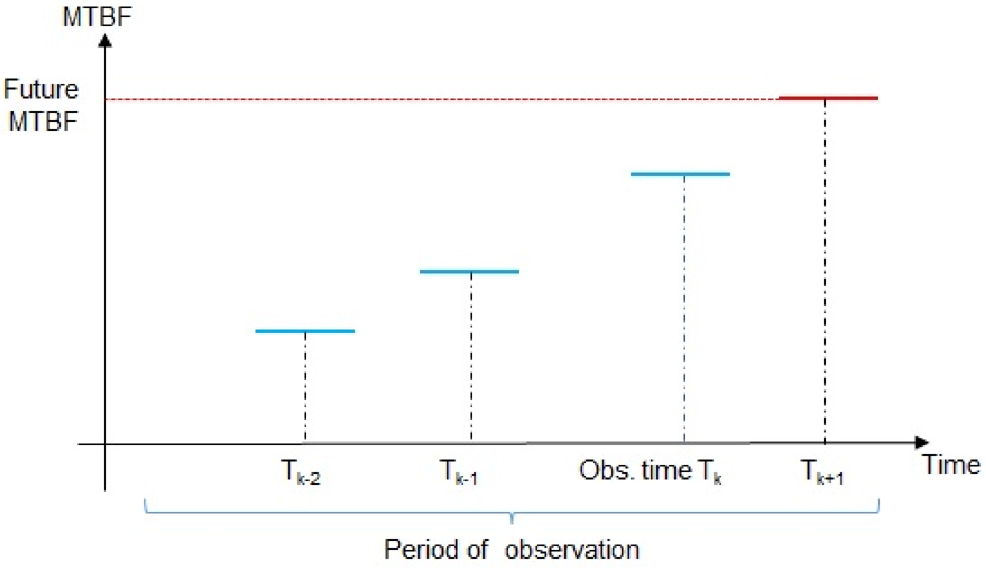

Figure C-3. Future MTBF

Figure C-4 shows the instantaneous MTBF. This value is observed at T_k without taking into account past values of MTBF.

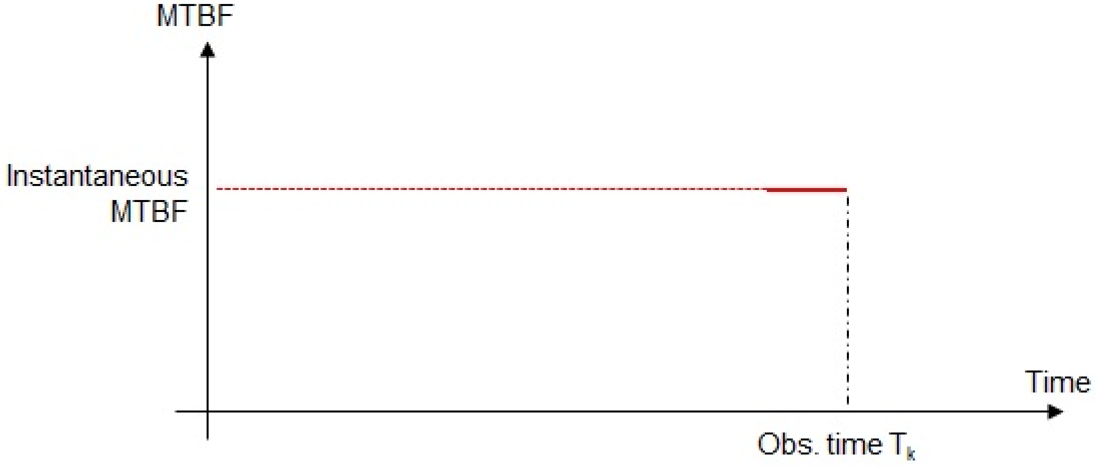

Figure C-4. Instantaneous MTBF

C.1.3 LINK BETWEEN MTTF AND THE RELIABILITY FUNCTION

The MTTF is mathematically expressed as $MTTF = E[T]$, where *T* is a random variable representing the time to failure of the system, and *E* is the expectation operator (see appendix B.1).

Using the definition of the expected value of *T*, the MTTF is related to the reliability function *R* of *T* by:

$$MTTF = \int_0^\infty tf(t)dt = \int_0^\infty R(t)dt \qquad \text{(C-1)}$$

C.2 STATISTICAL MODELS

Mathematical models are commonly used to describe a wide variety of phenomena. Statistical models are used when the complexity of the phenomena exceeds the current computational power (e.g., turbulence), or when variability and uncertainty plays a non-negligible role (e.g., pharmacokinetics). In addition to being used as descriptive means of the past and the present, statistical models serve as the basis for various extrapolations (e.g., animal and human responses to medications) and forecasting (e.g., reliability and airline revenue management).

Statistical models are of two types: parametric and non-parametric. A parametric model in statistics is a family of distribution functions that can be described by a finite set of parameters (e.g., atmospheric turbulence is modeled by a normal distribution defined by its mean and standard deviation). Parametric models are sought because their parameters capture structural information of the data, which allows:

- The application of data compression techniques (relieves computation burden).
- Insight into physical or other processes observed via the data, which can then be either applied to explaining past measurements or support predictions of future trends.
- The computation of quality of fit measures such as model-mismatch or model parameter quality measures.

The application of a parametric model implies that assumptions have been made on the format of the data, which may be construed as being too rigid. Non-parametric models require no, or little, assumptions to be made about the data, which gives them more flexibility and the following advantages over parametric models:

- Useful when dealing with unexpected or outlying data points that would be problematic with a parametric approach.
- Preferable when the assumptions required for a parametric model are not valid.

Their flexibility also causes a number of drawbacks, including:

- Difficulty modeling non-stationary processes because of the required larger number of computations and limited availability of software applications.
- Difficulty gaining insight into the physical or other processes underlying the data because of the lack of assumptions regarding the data within the model.
- All data must be kept (data processing might be expensive).
- Approximations of the probability density function are constructed from binning the collected data into empirical density functions, so that if data are collected online, probability density estimates often need to be recomputed using batch processing.

In biomedical statistics, parametric models are typically used to estimate effects (e.g., a test on a group of patients). Non-parametric models are used to perform hypothesis testing (e.g., to model the relative risk of developing a complication from a given health risk), as compared to a prescribed threshold (e.g., 50% chance of an outcome).

C.2.1 FORMULAE TO COMPUTE QUALITY METRICS FOR AN ESTIMATOR IN PARAMETRIC APPROACH

The estimator efficiency $Eff(\hat{\beta})$ is given as [C-1]:

$$Eff(\hat{\beta}) = \frac{\left[\frac{\partial}{\partial\beta} E(\hat{\beta})\right]}{I(\beta) Var(\hat{\beta})} \tag{C-2}$$

Where *I(β)* is the Fisher information function and is defined as:

$$I = Var\left[\frac{\partial}{\partial\beta} L(\beta)\right] \tag{C-3}$$

An estimator is efficient only if $Eff(\hat{\beta}) = 1$ or asymptotically efficient only if:

$$\lim_{n\to\infty} Eff(\hat{\beta}) = 1.$$

C.2.2 FORMULAE TO COMPUTE QUALITY METRICS FOR AN ESTIMATOR IN NON-PARAMETRIC APPROACH

The following different metrics can be applied to qualify the estimator.

The non-parametric form of the mean square error (MSE) is defined as:

$$MSE(\beta) = \frac{1}{n}\sum_{i=1}^{n}(\beta_i - \beta)^2 \tag{C-4}$$

The mean absolute value difference is defined as the average of the difference between the estimator value and the simulated values. That is:

$$MAVD(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left|\beta_i - \beta\right| \tag{C-5}$$

The magnitude relative error (MRE) is defined as the absolute value of the relative error. That is:

$$MRE(\beta) = \left|\frac{\hat{\beta} - \beta}{\beta}\right| \tag{C-6}$$

The mean magnitude of relative error (MMRE) is defined as the average value of the absolute value of the relative error. That is:

$$MMRE(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\beta_i - \beta}{\beta}\right| \tag{C-7}$$

The magnitude of error relative to the estimate (MER) is defined as the value of the error relative to the estimate. That is:

$$MER(\beta) = \left|\frac{\hat{\beta} - \beta}{\hat{\beta}}\right| \tag{C-8}$$

The mean magnitude of error relative to the estimate (MMER) is defined as the average value of MER. That is:

$$MMER(\beta) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{\beta}_i - \beta}{\hat{\beta}_i}\right| \tag{C-9}$$

The median of absolute residual is obtained from the values of the absolute residual, defined as:

$$AR(\beta) = \left|\hat{\beta} - \beta\right|. \tag{C-10}$$

C.2.3 PARAMETRIC MODEL FITTING USING THE METHOD OF MOMENTS

The method of moments is a technique for constructing estimators of the parameters describing the statistical model based on matching the moments of the sample data with the corresponding moments of the model. From this definition, the number of sample moments needs to be equal to the number of parameters to be estimated: To estimate one parameter only, the first moment is necessary; to estimate two parameters, both the first and the second moments need to be computed.

For a random variable *T*, the moments are defined as:

$$\mu_n = E\left(T^n\right) \tag{C-11}$$

These moments are related to the distribution function of T and therefore to its parameter θ. If, for example, there are l parameters to be estimated, the method of moments has three steps:

1. Define the l first moments of T: μ_k with $k=1,\ldots l$.
2. Compute the corresponding l moments on the sample data:

$$m_k = \frac{1}{n}\sum_{i=1}^{n} f_\theta^k(t_i) \tag{C-12}$$

3. Solve the system of equations given by:

$$\mu_k = m_k \tag{C-13}$$

The advantage of the method of moments is that it often provides estimators when other methods fail or when estimators are hard to obtain (e.g., Gamma distribution). The drawback is that they are usually not the best estimators (in an MSE sense).

C.2.4 PARAMETRIC MODEL FITTING BY MAXIMIZING THE LIKELIHOOD FUNCTION

The function:

$$L_t(\theta) \overset{\Delta}{=} f_\theta(t) \tag{C-14}$$

is the likelihood of θ (equivalent of the model f_θ, which θ parameterizes) given the measured data t. The maximum likelihood estimator (MLE) method is based on the fact that maximizing the likelihood function is equivalent to maximizing the logarithm of the likelihood function[4]. This method is valid for both discrete and continuous data.

For example, in the case of continuous data composed of i independent samples of observations t, the likelihood function is given by:

$$L_t(\theta) = \prod_{i=1}^{n} f_\theta(t_i) \tag{C-15}$$

And the logarithm of the likelihood function is defined as:

$$\Lambda_t(\theta) = \ln\left[L_t(\theta)\right] = \sum_{i=1}^{n} \ln[f_\theta(t_i)] \tag{C-16}$$

[4] In the literature, the MLE method can also be described as minimizing the negative logarithm of the likelihood function. Because the logarithm function is a strictly increasing function, this is equivalent.

Maximizing $\Lambda_t(\theta)$ is then transformed into equating its derivative to zero and leads to the following system of equations or "scores" functions:

$$\sum_{i=1}^{n}\left[\frac{\partial}{\partial\theta_i}\left(\ln(f_\theta(t_i))\right)\right]=0 \tag{C-17}$$

The system of equations is finally solved for the parameters θ_i.

If the mathematical model $f_\theta(.)$ is a linear Gaussian model, the inverse problem is linear and can be explicitly solved. When the inverse problem has no explicit solution, numerical methods such as Newton-Raphson can be used.

When the sample size is large (e.g., over 30 samples), the MLE is unbiased, consistent, normally distributed, and efficient. The drawbacks, however, include the possibility of MLE being highly biased for small samples, and its sensitivity to initial conditions may generate a local optimum, not a global optimum (this is common when the number of parameters to be estimated is large).

C.2.5 PARAMETRIC MODEL FITTING BY LEAST SQUARE ESTIMATION

The method of least squares allows the estimation of parameters by minimizing the squared discrepancies (also called residuals) between observed data and their expected values. In the context of lifetime data, the problem is termed a regression problem, in which the variation in one variable (i.e., the response variable) can be partly explained by the variation in the other variables (i.e., covariates). For example, the variation in survival times can be primarily explained by the variation in environmental conditions [C-2].

Mathematically, the data sample $T=\{t_i\}$ can be expressed as:

$$T = f_\theta(X)+\eta\,, \tag{C-18}$$

where $f_\theta(.)$, called a regression function defined by parameter(s) θ, $X=\{x_i\}$, denotes the covariates, and η captures the noise in the data. The method of least squares is a standard approach to the approximate solution of overdetermined sets of equations in which there are more equations (values of response variables and covariates) than unknown parameters θ. The least square estimator $\hat{\theta}$ is the value of θ that minimizes:

$$\sum_{i=1}^{n}[t_i - f_\theta(x_i)]^2 \tag{C-19}$$

The least square criterion above is a computationally convenient measure of fit. When the noise is normally distributed with equal variances, the least square estimator (LSE) corresponds to the MLE. When the noise's variance σ_η is dependent on the covariates (i.e., observations are more or less accurate), the weighted LSE provides a useful extension as:

$$\sum_{i=1}^{n} \frac{[t_i - f_\theta(x_i)]^2}{\sigma_{\eta_i}^2} \tag{C-20}$$

A commonly used weighted LSE in the context of density estimation is the minimum chi-squared (or χ^2) estimator.

When the *f (.)* is a linear function of the parameters θ, the method is called linear regression and can be solved in close-form. When *f (.)* is a nonlinear function of the parameters θ, nonlinear regression resorts to iterative algorithms to compute the LSE.

Lastly, regression can also be applied to non-parametric models by assuming a relaxation of some quantitative assumptions on *f (.),* such as monotonicity or smoothness [C-3].

C.3 RELIABILITY MODELS

The following sections list models that are currently used in safety applications for hardware/software depending on their ability to fit the data and possess applicable assumptions. The models are categorized into discrete-time models and continuous-time models.

C.3.1 DISCRETE-TIME MODELS

The absence of wear out for software can be related to the natural assumption that the failure intensity remains constant as long as the software does not change. That is, times between failures can be modeled as a random variable with an exponential distribution. The exponential time between failures is a model in which times between two consecutives failures are independent random variables with exponential distributions.

The failure intensity has the shape versus time shown in figure C-5.

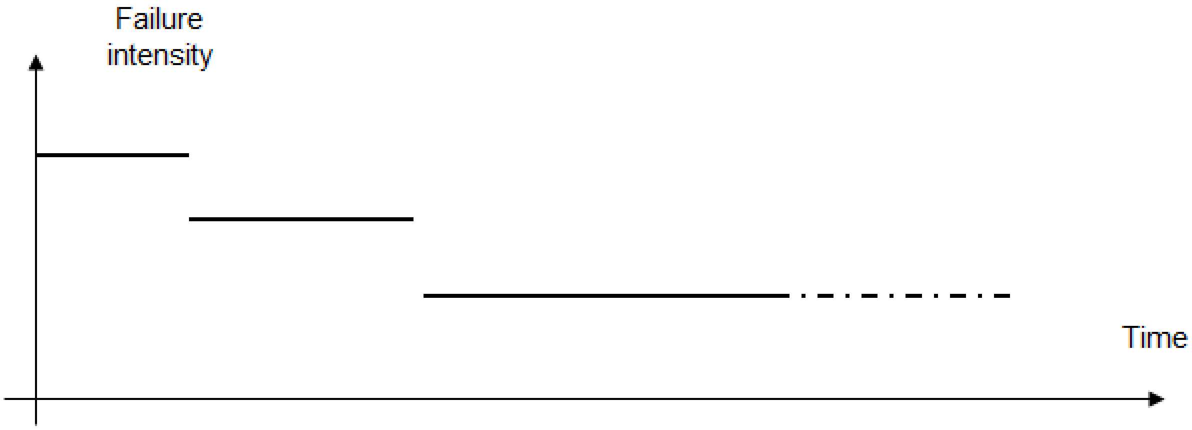

Figure C-5. Shape of failure intensity function over time

C.3.1.1 Jelinski-Moranda Model

The Jelinski-Moranda (JM) model was introduced in 1972 and is often referred to as the first model of software reliability [C-4]. The times between software failures are assumed to be statistically independent exponential random variables.

The following assumptions are carried with the use of the model:

- Initially, the software contains an unknown number, *N*, of faults.
- When a failure occurs, the fault that caused it is eliminated perfectly, and no new fault is introduced into the software.
- Each fault still presents the same contribution to the failure intensity (i.e., each fault has the same potential for instantaneously causing a failure).
- The failures are not correlated (i.e., the times between failures are independent).
- The failure intensity is proportional (with a coefficient Φ) with the number of residual faults.

From the above assumptions, the failure intensity, λ, can be modeled as:

- At the beginning, *N* faults are present: $\lambda = \Phi N$.
- After the first failure, *N-1* faults remain: $\lambda_1 = \Phi(N-1)$.
- After the second failure, *N-2* faults remain: $\lambda_2 = \Phi(N-2)$, and so on.

So the failure intensity for the *i*th occurring failure, $\lambda_{I,}$ is given by:

$$\lambda_i = \Phi(N - i + 1) \tag{C-21}$$

where the constant Φ represents the occurrence rate.

Figure C-6 shows the failure intensity behavior versus time. Using the information from each test, the MLE is mostly used to estimate the model parameters; in particular, the number of faults initially present, *N*, are of interest.

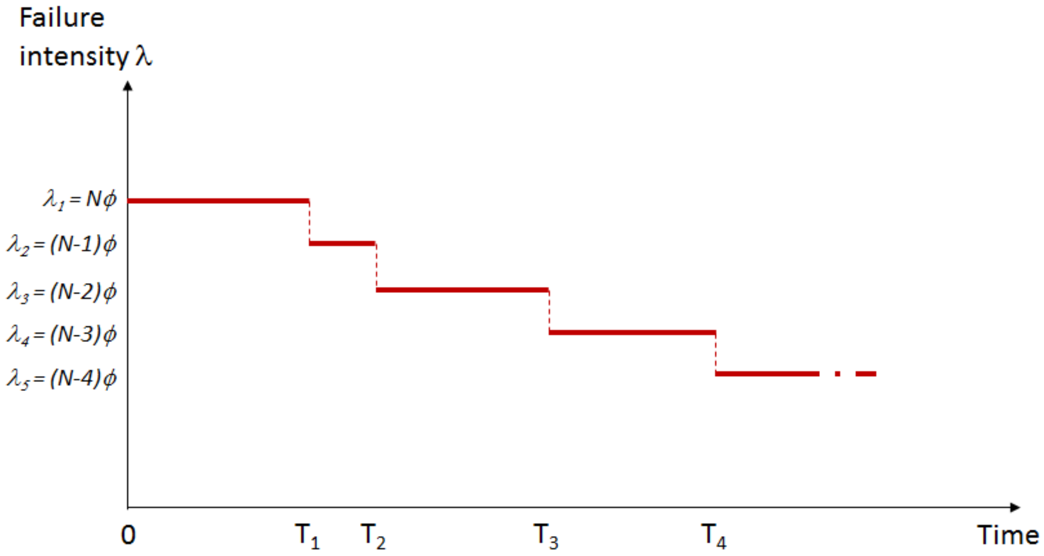

Figure C-6. Failure intensity in JM model

This model is perhaps not the most realistic model for computer systems, but it was among the first ones and may be used as a starting point. Among its numerous disadvantages are the following [C-1]:

- When the initial *N* faults are corrected, the software is perfect, which is not plausible.
- The faults have the same rate of event. In fact, certain faults occur very quickly and others only after a long period of use of the software.
- The choice of a number of initial faults as the template parameter is questionable. What matters to the user of the software is not the total number of mistakes but the frequency of occurrence of failures.
- The model assumes that all faults have the same severity, though in real life, some faults are benign and others significant; faults can occur at each execution instruction or once in a million.
- The estimation of the parameters by the method of maximum likelihood is not easy and is not of good quality.
- Any activated fault is supposed to be completely spotted and perfectly corrected, which is not necessarily the case. Therefore, the JM model is considered to overestimate reliability and dubbed “optimistic.”

C.3.1.2 Geometric Moranda Model

Moranda proposed a modification to the JM model, termed the geometric de-eutrophication model, or Geometric Moranda (GM) model [C-5]. In this modification, the failure intensity decreases geometrically with the detection (and fixing) of a fault.

The failure intensity is given by:

$$\lambda_i = \exp(-c)\lambda_{i-1} \tag{C-22}$$

Figure C-7 shows the failure intensity behavior versus time.

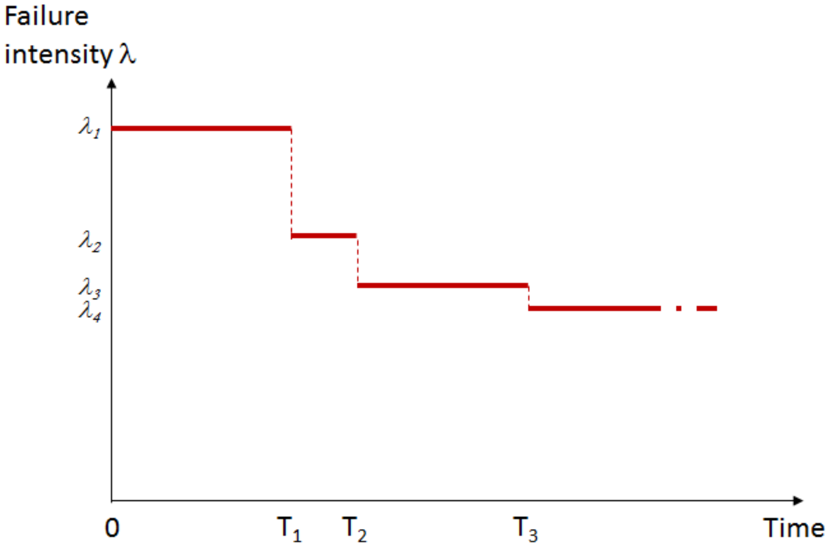

Figure C-7. Failure intensity in GM model

The estimation of the parameters by the method of maximum likelihood needs a numerical method, such as a recursive algorithm, which is fairly easy to implement [C-6]. The main shortcoming of the GM model resides in the assumption that all faults have the same likelihood of occurrence. In fact, faults have a different activation potential, and the faults with high potential for activation tend to occur very early, and their correction will greatly reduce the failure intensity. The improvement due to corrections must therefore logically be strong at the beginning and then be smaller and smaller.

C.3.1.3 Discrete-Time Semi-Markov Chain

Discrete-time semi-Markov chains are less studied than the continuous-time type (see section C.3.2), though they are useful in modeling renewal processes in repairable systems based on time-truncated data [C-7]. Figure C-8 shows the evolution in time of such a discrete-time semi-Markov chain.

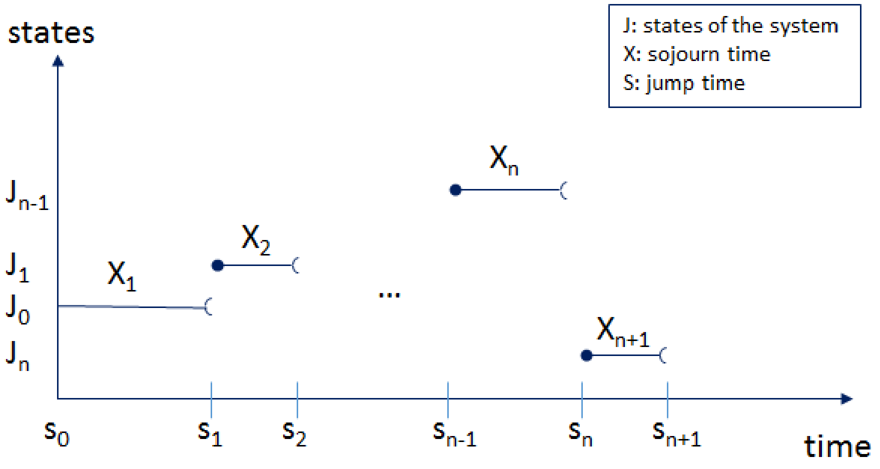

Figure C-8. Sketch of sample path for Markov renewal chain

Markov chains are described by a finite state space, *S={s}*; a transition rate matrix, $Q=[q_{ij}]$, whose elements describe the rate at which the system transitions from state *i* to state *j*; and the initial probability distribution defined on the state space. The model is termed as semi-Markov because it is not entirely memoryless (or Markovian) like its continuous-time counterpart; rather, the Markovian property only applies to the jump times.

The reliability function, *R(t)*, of a discrete-time semi-Markov system at a time *t* is the probability that the system has functioned without failure in the period *[0,t]*. The state space of such a system is split into the subset of working states (or up states) denoted by *U* and the subset of failure states (or down states) denoted by *D*.

$$\begin{aligned} U &= \{1, \cdots, s_1\} \\ D &= \{s_1 + 1, \cdots, s\} \end{aligned} \tag{C-23}$$

Using the state space partitioning property, the reliability function can be expressed as:

$$R(t) = \alpha_1 \left[\delta I - q_{11}\right]^{-1} \left[I - diag(Q.1)_{11}\right](t) 1_{s_1} \tag{C-24}$$

where α_1 represents the first element of the row vector of the initial distribution of the semi-Markov chain, q_{11} is the first element of the discrete-time semi-Markov kernel matrix Q, and $1_{\{i=j\}}(k)$ is equal to 1 if *i=j* and k positive and 0 elsewhere.

The failure rate was introduced in 1963, by Barlow, Marshall, and Prochan, and bears their names as the "BMP-failure rate:"

$$\lambda(t) = \begin{cases} 1 - R(0) & t = 0 \\ 1 - \dfrac{R(t)}{R(t-1)} & R(t-1) \neq 0 \\ 0 & otherwise \end{cases} \tag{C-25}$$

Note that the failure rate in the discrete time case is not a general positive function as in the continuous case of semi-Markov processes.

For reliability analysis, two mean times are of interest: the MTTF and the mean time to repair. Both can be expressed as hitting times into the state space *D* and state space *U*, respectively:

$$\begin{aligned} MTTF &= \alpha_1 (I - p_{11})^{-1} m_1 \\ MTTR &= \alpha_2 (I - p_{22})^{-1} m_2 \end{aligned} \tag{C-26}$$

where p_{11} and p_{12} are the diagonal elements of the transition matrix, *P*, associated with the semi-Markov chain [C-8], and *m* is the vector of mean sojourn times in up states (m_1) and down states (m_2).

Reliability estimators for the discrete-time semi-Markov process are based on the MLE, which can be applied on the process's kernel *Q* and transition function *P* to derive the estimator for the reliability function, *R*.

The MLE of the reliability function of a discrete-time semi-Markov chain is consistent and asymptotically normal. Because the failure rate is defined in terms of the reliability function, the same properties apply to the failure rate estimator. Confidence intervals are further defined in [C-8].

C.3.2 CONTINUOUS-TIME MODELS

Random jump processes can be generalized under the name of Markov renewal processes (MRP). Special cases of MRP cover Markov chains, Poisson processes, and renewal processes. Because of the vastness of the topic, this paper focuses on Poisson processes as they have been widely used in reliability engineering and offer a sideline on Markov chains for their use mostly in queuing theory.

C.3.2.1 Poisson Processes

A Poisson process is a simple and widely used stochastic process for modeling the times at which arrivals enter a system; the term "arrival" is used to represent the time at which some repeating event (e.g., failure) occurs. In this process, arrivals may occur at arbitrary positive times. Any arrival process can be equivalently specified in terms of the joint distribution of the arrival times, inter-arrival intervals, or random variables counting the arrivals in given intervals.

For a Poisson process, using the inter-arrival intervals is the most convenient way to specify the process; these intervals have an exponential distribution function [C-9].

Poisson processes stand apart from other renewal processes because of the memoryless property of the exponential distribution, which is appropriate for catalectic failures.

The most commonly used models in the reliability analysis of software are the non-homogeneous Poisson processes (NHPP), which are based on the assumption that the faults that have been incorporated in the software during the development phase are removed during the testing phase. Therefore, there is no jump after each failure, and this model is suitable to model minimal maintenance (see figure C-9). More details on NHPP are provided in section C.3.2.1.2.

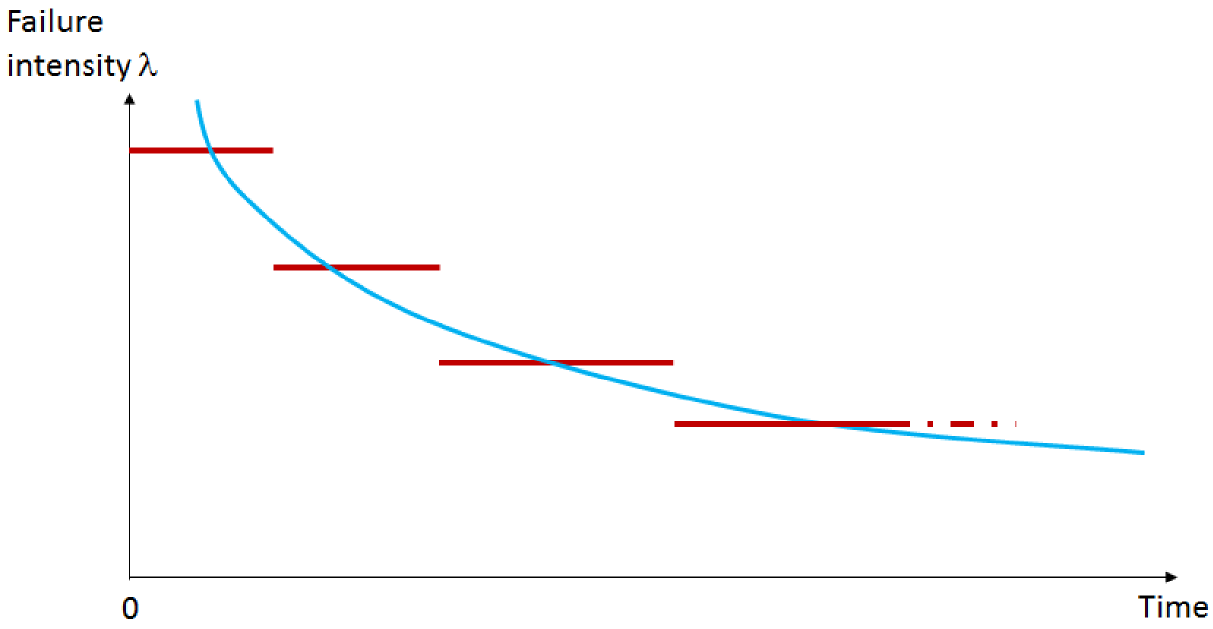

Figure C-9. Failure intensity in a NHPP model

The MLE for the three types of data is given by:

$L = \prod_{i=1}^{n} \lambda(t_i) \exp(-m(t_n))$	for failure truncated data
$L = \prod_{i=1}^{n} \lambda(t_i) \exp(-m(t))$	for time truncated data
$L = \exp(-m(\tau_k)) \prod_{j=1}^{k} \frac{(m(\tau_j) - m(\tau_{j-1}))^{n_j}}{n_j!}$	for grouped data

(C-27)

where n is the number of failures, t_i the time of the i^{th} failure, k the number of groups, and $[\tau_k, \tau_{k-1}]$ the time interval of the k^{th} group.

C.3.2.1.1 Homogeneous Poisson Process Model

The Homogeneous Poisson Process (HPP) is a counting process where the failure intensity is a time and deterministic function of time defined as:

$$\begin{aligned} \lambda(t) &= \lambda \\ m(t) &= \lambda t \end{aligned} \qquad \text{(C-28)}$$

For the three types of data, the MLE is given by:

$L = \lambda^n \exp(-\lambda t_n)$	for failure truncated data
$L = \lambda^n \exp(-\lambda t)$	for time truncated data
$L = \exp(-m(\tau_k))\lambda^n \prod_{j=1}^{k} \frac{(\tau_j - \tau_{j-1})^{n_j}}{n_j!}$	for grouped data

(C-29)

For the three types of data, the parameter estimator is given by:

$\hat{\lambda} = \frac{n}{t_n}$	for failure truncated data
$\hat{\lambda} = \frac{n_t}{t}$	for time truncated data
$\hat{\lambda} = \frac{n_{\tau_k}}{\tau_k}$	for grouped data

(C-30)

The cumulated, instantaneous, and future MTBF are equal to:

$$MTB\hat{F} = \frac{1}{\hat{\lambda}} \tag{C-31}$$

For the three types of data, the statistical qualities of the estimator are defined by:

The failure intensity is biased, Unbiased estimator: $\hat{\lambda}' = \frac{n\lambda}{n-1}$ Fisher information: $I(\lambda) = \frac{n}{\lambda^2}$ Efficiency: $Eff(\lambda') = \frac{n-2}{n}$ (asymptotically efficient)	for failure truncated data
The estimator is unbiased and efficient	for time truncated data
Same as for time truncated data	for grouped data

(C-32)

The confidence bounds on λ at risk level α are given by:

$\left[\dfrac{\chi_2^{-1}\left(\dfrac{\alpha}{2};2n\right)}{2T_n};\dfrac{\chi_2^{-1}\left(1-\dfrac{\alpha}{2};2n\right)}{2T_n}\right]$	for failure truncated data
$\left[\dfrac{\chi_2^{-1}\left(\dfrac{\alpha}{2};2N_t\right)}{2t};\dfrac{\chi_2^{-1}\left(1-\dfrac{\alpha}{2};2(N_t+1)\right)}{2t}\right]$	for time truncated data
Same as for time truncated data	for grouped data

(C-33)

where χ_2^{-1} is the first quantile of the chi-square distribution.

The HPP model can only be used when no corrections have been implemented because it assumes a constant failure rate (exponential distribution). Unfortunately, this is never the case in repairable systems and, therefore, NHPP process models have been successfully used in reliability engineering.

C.3.2.1.2 NHPP Models

As a general class of well-developed stochastic process models in reliability engineering, NHPP models have been successfully used in studying hardware reliability problems. In these models, the number of failures experienced up to time *t* follows an NHPP distribution.

The NHPP model class is a close relative of the HPP model, with the difference that the expected number of failures is allowed to vary with time. Therefore, they are useful for both calendar time data and for the execution time data. As indicated in the second bulleted assumption below, NHPP assumes that the more faults are detected up to time *t*, the more undetected faults can be expected to still exist in the software; that relationship is proportional, which supports comparisons between systems. For example, if over a time period, *T*, the number of faults detected/removed in System A is 10 times higher than for System B, it can be expected that System A still contains 10 times more faults than System B.

Other important advantages of NHPP models that should be highlighted are that NHPPs are closed under superposition and time transformation. Two or more existing NHPP models can easily be incorporated by summing up the corresponding mean value functions. The failure intensity of the superposed processes is also the sum of the failure intensities of the underlying processes.

Some of the basic assumptions (apart from some special ones for the specific models discussed below) assumed for NHPP models are as follows:

- A software system is subject to failure during execution caused by faults remaining in the system.
- The number of faults detected at any time is proportional to the remaining number of faults in the software.
- The failure rate of the software is equally affected by faults remaining in the software.
- Once a failure is detected, repair efforts start and the fault causing the failure is removed with certainty.
- All faults are mutually independent from a failure detection point of view.
- The proportionality of failure occurrence/fault isolation/fault removal is constant.
- For each fault detection/removal phenomenon at the manufacturer/user end, there exists an equivalent fault detection/fault removal at the user/manufacturer end.
- The fault detection/removal phenomenon is modeled by NHPP.

One of the most used NHPP models is the Power Law Process (PLP). A key reason is that the model parameters have a closed form expression so that no numerical method is needed. As a direct consequence, the statistical properties of its estimated parameters can be easily expressed.

The failure intensity for the PLP model is defined as:

$$\lambda(t) = \lambda\beta t^{\beta-1}. \qquad \text{(C-34)}$$

For the three types of data, the MLE is given by:

$L = \lambda^n \beta^n \left[\coprod_{i=1}^{n} t_i^{\beta-1} \right] \exp(\lambda t_n^{\beta})$	for failure truncated data
$L = \lambda^n \beta^n \left[\coprod_{i=1}^{n} t_i^{\beta-1} \right] \exp(\lambda t^{\beta})$	for time truncated data
$L = \exp(-\lambda \tau_k^{\beta}) \lambda^n \coprod_{i=1}^{k} \frac{(\tau_i^{\beta} - \tau_{i-1}^{\beta})^{n_j}}{n_j!}$	for grouped data

(C-35)

where n is the number of failures, t_n the time of the n^{th} failure, and k the number of groups.

For the three types of data, the parameter estimator is given by:

$\hat{\beta} = \dfrac{n}{\sum_{i=1}^{n} \ln\left(\dfrac{T_n}{T_i}\right)}; \hat{\alpha} = \dfrac{n}{T_n \hat{\beta}}$	for failure truncated data
$\hat{\beta} = \dfrac{n_t}{\sum_{i=1}^{n_i} \ln\left(\dfrac{t}{T_i}\right)}; \hat{\alpha} = \dfrac{n_t}{T_n \hat{\beta}}$	for time truncated data
$\hat{\alpha} = \dfrac{n_{\tau_k t}}{\tau_k \hat{\beta}}; \sum_{i=1}^{k} \left[\left(n_{\tau_j} - n_{\tau_{j-1}} \right) \left[\dfrac{\tau_j^{\beta} \ln(\tau_j) - \tau_{j-1}^{\beta} \ln(\tau_{j-1})}{\tau_j^{\beta} - \tau_{j-1}^{\beta}} - \ln(\tau_k) \right] \right]$ Unlike in the previous case, the estimator has no closed form expression.	for grouped data

(C-36)

The three types of MTBF are:

The cumulated MTBF is given by	$M\hat{T}BF_c = \dfrac{1}{\lambda t^{\beta-1}}$
The instantaneous MTBF is given by	$M\hat{T}BF_i = \dfrac{1}{\lambda \beta t^{\beta-1}}$
The future MTBF is given by	$M\hat{T}BF_f = \exp(\lambda t^{\beta})\Gamma\left(1 + \dfrac{1}{\beta}\right)\lambda^{-\frac{1}{\beta}}\left[1 + F_{G\left(\frac{1}{\beta};\lambda\right)}(t^{\beta})\right]$

(C-37)

where $F_G(t)$ is the cumulated probability density function of the gamma distribution.

For the three types of data, the estimator statistical qualities are given by:

The β parameter is biased, Unbiased estimator: $\hat{\beta}' = \dfrac{n\beta}{n-2}$	for failure truncated data
The β parameter is biased, Unbiased estimator: $\hat{\beta}' = \dfrac{n\beta}{n-1}$	for time truncated data
No closed form expression exists	for grouped data

(C-38)

The confidence bounds on λ, at risk level α, are given by:

$\left[\frac{\beta\chi_2^{-1}\left(\frac{\alpha}{2};2(n-1)\right)}{2n};\frac{\beta\chi_2^{-1}\left(1-\frac{\alpha}{2};2(n-1)\right)}{2n}\right]$	for failure truncated data
$\left[\frac{\beta\chi_2^{-1}\left(\frac{\alpha}{2};2n\right)}{2n};\frac{\beta\chi_2^{-1}\left(1-\frac{\alpha}{2};2n\right)}{2n}\right]$	for time truncated data
Same as for time truncated data	for grouped data

(C-39)

These estimators have an explicit expression, which is rare in software reliability. This explains, in large part, the popularity of this model.

C.3.2.2 Generalized Power Family Models

The Generalized power family (GPF) models are a generalization of the PLP models defined as:

$$\begin{aligned}\lambda(t) &= \lambda\beta k(t)^{\beta-1}\frac{\partial}{\partial t}k(t)\\ m(t) &= \lambda k(t)^{\beta}\end{aligned} \tag{C-40}$$

So that:

- For the PLP model, *k(t)* = 1.
- For the log-power (LP) model:

$$m(t)=\ln(1+t);\lambda(t)=\frac{\lambda\beta\left[\ln(1+t)^{\beta-1}\right]}{1+t} \tag{C-41}$$

- For the general LP model:

$$m(t)=\ln(1+\ln(1+t));\lambda(t)=\frac{\lambda\beta\left[\ln(1+\ln(1+t))\right]^{\beta-1}}{(1+t)(1+\ln(1+t))} \tag{C-42}$$

The GPF family of models is large enough to accommodate a wide range of actual data sets, which makes it particularly attractive. For some reliability data sets, the PLP model is ill-suited because reliability increases too fast due to the $t^{\beta-1}$ term in the failure intensity function. The *k(.)* function is thus selected to adapt to the increase. The study performed in [C-10] shows that several GPF models were better suited to the data than the PLP model.

C.3.2.3 Modified Power Law Process

The Modified Power Law Process is an NHPP for which $k(t) = \lambda t + \gamma t^{\beta}$ and the failure intensity is given by:

$$\lambda(t) = \lambda + \gamma\beta t^{\beta-1}; \lambda(0) = +\infty \tag{C-43}$$

This model is a superposition of two Poisson processes:

- An HPP process with failure intensity λ.
- A PLP process.

The advantage of this model is that the failure intensity has a limit as $t \to +\infty$:

$$\lim_{t\to+\infty} \lambda(t) = \lambda \tag{C-44}$$

C.3.2.4 Doubly Bounded Power Law Process

The Doubly Bounded Power Law Process (DBPLP) is an NHPP where the failure intensity is defined as:

$$\lambda(t) = \lambda + \frac{\gamma\beta}{(\exp(t) + t)^{1-\beta}} \tag{C-45}$$

The advantage of this model is that the failure intensity has a limit at both *t* = 0 and as $t \to +\infty$:

$$\lambda(0) = \lambda + \beta\gamma; \lim_{t\to+\infty} \lambda(t) = \lambda \tag{C-46}$$

C.3.2.5 Hyper-Exponential Model

The Hyper-Exponential model is an NHPP obtained from assuming that the first moment of a failure is a two-component hyper geometric distribution. Its failure intensity is defined as:

$$\lambda(t) = \frac{p\lambda_1 \exp(-\lambda_1 t) + (1-p)\lambda_2 \exp(-\lambda_2 t)}{p \exp(-\lambda_1 t) + (1-p) \exp(-\lambda_2 t)} \tag{C-47}$$

A remarkable property of this model is that the failure intensity tends toward a non-zero limit as $t \to +\infty$. It is one of the few models (together with the DBPLP) to present this realistic feature.

C.3.2.6 Littlewood Model

The Littlewood model is an NHPP with $k(t) = -N\beta^{\alpha}(\beta + t)^{-\alpha}$ and failure intensity defined as:

$$\lambda(t) = \frac{N\alpha\beta^{\alpha}}{(t+\beta)^{\alpha+1}} \qquad \text{(C-48)}$$

C.3.2.7 Goel-Okumoto Model (Derivative of the JM Model)

The Goel-Okumoto (GO) model is based on the following assumptions:

- The software contains, at the initial instant, a random fault, *N*, with expectation *a*.
- When a failure occurs, the fault that caused it is perfectly eliminated.
- No new fault is introduced.
- The failure intensity is proportional to the mean number of residual faults. The proportionality coefficient is denoted *b* and can be interpreted as the rate of manifestation of the faults.

The failure intensity is defined as:

$$\lambda(t) = ab.\exp(-bt)\,. \qquad \text{(C-49)}$$

This model is similar to the JM model, except for two points:

- The initial number of faults is a random variable and no longer an unknown constant.
- The failure intensity is assumed to be proportional to the average number of residual faults and no longer an absolute number.

C.3.2.8 Musa-Okumoto Model (Derivative of the JM Model)

The Musa-Okumoto model is derived from the JM model and is an approximation of an NHPP. Its failure intensity is given by:

$$\lambda(t) = \frac{\lambda}{1+\lambda ct} \qquad \text{(C-50)}$$

C.3.2.9 Ohba Model (Derivative of the JM Model)

The Ohba (O) model is based on the following assumptions:

- The software contains, at the initial instant, a random fault, *N*, with expectation *a*.
- When a failure occurs, the fault that caused it is perfectly eliminated.
- No new fault is introduced.
- The rate of manifestations of the faults denoted *b* is not constant but depends on time.

The failure intensity is defined as:

$$\lambda(t) = \frac{ab(1+\beta)\exp(-bt)}{1+\beta\exp(-bt)} \qquad \text{(C-51)}$$

C.3.2.10 Yamada-Ohba-Osaki Model (S-Shaped Model)

The appellation of Yamada-Ohba-Osaki (YOO) collects several models also known as S-shaped models [C-11]. The YOO1 model is based on the same assumptions as the O model but defines the parameter *b* as:

$$b(t) = \frac{b^2 t}{1+bt} \qquad \text{(C-52)}$$

The failure intensity is defined as:

$$\lambda(t) = ab^2 t\exp(-bt) \qquad \text{(C-53)}$$

The YOO2 model is based on the same assumptions as the YOO1 model but introduces the notion that, in practice, the correction of a fault is likely to be imperfect. This imperfection translates in the expectation *a* parameter, which is no longer constant but varies with time as $a(t) = a\exp(\alpha t)$.

The failure intensity is defined as:

$$\lambda(t) = \frac{ab}{b+\alpha}\left[\alpha\exp(-\alpha t) - b\exp(-bt)\right] \qquad \text{(C-54)}$$

C.3.2.11 Khoshgoftaar Model (S-Shape Model)

The Khoshgoftaar (KHO) model is a generalization of the GO and YOO models. It is based on the “k-Erlangian” model, for which the failure intensity is defined as:

$$\lambda(t) = \frac{ab^{k+1}t^k}{k!}\exp(-bt) \qquad \text{(C-55)}$$

When k=0, the KHO model turns into the GO model and, for k=1, the YOO model.

C.3.2.12 Logistic Growth Curve model (S-Shape Model)

The logistic growth curve model has an S-shape and is characterized by:

$$\begin{aligned} m(t) &= \frac{a}{1+k\exp(-bt)} \\ \lambda(t) &= \frac{ab\exp(-bt)}{(1+k\exp(-bt))^2} \end{aligned} \qquad \text{(C-56)}$$

C.3.2.13 Continuous-Time Markov Chains

In a continuous-time Markov chain (CTMC), the values taken by the model are finite and countable, and the time spent in each state is a non-negative real value with an exponential distribution. CTMCs exhibit the so-called Markov property that the future behavior, both in terms of the remaining time in the current state and the next state, solely depends on the current state of the model and not on any of the previous (historical) states.

CTMCs are widely used to model software systems. They are typically generated from high-level specifications, prior to the development of the system, and used for quantitative evaluation of reliability and performance (e.g., throughput of production lines, average failure times) of complex systems such as queuing networks or stochastic Petri Nets. For queuing networks, CTMCs can be used where the upward transitions (e.g., job arrivals) occur at a rate that follows a Poisson process and where downward transitions (e.g., completed services leaving the queue) occur at a rate exponentially distributed.

C.4 REFERENCES

C-1 Gaudoin, O. and Ledoux, J., “Modélisation Aléatoire de la fiabilité des logiciels” [Stochastic Model for Software Reliability], Hermes, 2007.

C-2 van de Geer, S.A., “Least Squares Estimation,” *Encyclopedia of Statistics in Behavioral Science*, Vol. 2, Wiley, 2005.

C-3 Robertson, T. et al., *Order Restricted Statistical Inference*, Wiley, 1988.

C-4 Jelinski, Z. and Moranda, P., “Software Reliability Research,” *Statistical Computer Performance Evaluation*, Academic Press, 1972.

C-5 Moranda, P., “Prediction of Software Reliability and its Applications,” *Annual Reliability and Maintainability Symposium (RAMS),* 1975.

C-6 Vasanthi, T. and Arulmozhi, G., “Reliability Computation of Moranda’s Geometric Software Reliability Model,” *Economics Quality Control*, Vol. 22, No. 2, 2007.

C-7 Port, S.C., *Theoretical Probability for Applications*, Wiley, 1994.

C-8 Barbu, V.S. and Limnios, N., “Reliability of Semi-Markov Systems in Discrete Time: Modeling and Estimation,” 2006.

C-9 MIT course material 6.262, “Poisson Processes,” Discrete Stochastic Processes, Chapter 2, available from MIT open courseware at ocw.mit.edu (accessed on 01/11/2016).

C-10 Arnoux, F. et al., “The Generalized Power Family in Software Reliability Data Analysis,” *2nd International Conference on Mathematical Methods in Reliability (MMR),* 2000.

C-11 Yamada, S. and Osaki, A., “S-Shaped Reliability Growth Modeling for the Software Error Detection,” IEEE Transactions on Reliability, 1983.

APPENDIX D—EXAMPLE CONTENT OF IN-SERVICE REPORT

D.1 FIELD RETURN REPORT

The field return report is a description of the product in terms of carrier, product or equipment, or board characteristics, as detailed in table D-1.

Table D-1. Data elements recommended for inclusion in a field return report

Topic	Data Elements
Header	Report reference number and date (initial issue); author; reference document number and issue date; and origin (center).
Aircraft	Designation, make/model, date of entry into service, cumulated operating hours, and cumulated flight hours.
Equipment	Designation, reference, serial/part number, date of entry into service, cumulated operating hours (if different from aircraft), cumulated flight hours (if different from aircraft), and any comment relevant to equipment's history.
Rack/Cabinet	Designation, reference, serial/part number, date of entry into service, cumulated operating hours (if different from aircraft/equipment), cumulated flight hours (if different from aircraft/equipment), and any comment relevant to cabinet's history.
Board	Designation, reference, serial/part number, date of entry into service, cumulated operating hours (if different from equipment/cabinet), cumulated flight hours (if different from equipment/cabinet), and any comment relevant to board's history.

D.2 MAINTENANCE ACTION REPORT

The maintenance action report is a description of the maintenance actions performed on the failed product, as detailed in table D-2.

Table D-2. Data elements recommended for inclusion in a maintenance action report

Topic	Data Elements
Header	Report reference number and date (initial issue); author; expeditor (company or department); date on which the material was taken charge of and responsible party (entity/repairperson); and reference document for the repair.
Designation of the material to be repaired	Information of higher-level elements as appropriate (e.g., cabinet, equipment)–Designation, reference, and part number. Information on the element to be repaired–Type (e.g., cabinet/rack, equipment, board), designation, reference, and part number.
Description of the issue/failure	Failure confirmation (yes/no), symptoms, type of issue (e.g., degraded performance, and intermittent/permanent/induced/latent/ recurrent failure).
Miscellaneous	Observations–Traces of burnout on the box or the card, type of testing performed on the material, and observation after opening of the box.
Information related to the dismantling of the material	Before taking it apart–Has a visual examination been performed (yes/no)? How was the component dried? What technique was used to remove the glue? Method for taking the material apart. Number of components to be assessed. Dismantling information–Topologic marker, designation, manufacturer, reference of the dismantled material, and request for further assessment (yes/no).

D.3. FAILURE ANALYSIS REPORT

The failure analysis report is a description of the failure analysis on failed components, as shown in table D-3.

Table D-3. Data elements recommended for inclusion in a failure analysis report

Topic	Data Elements.
Header	Report reference number and date (initial issue), author, date of start of failure analysis, expert name/company, internal reference for the assessment request, and number of components to be assessed
Identification	Family/subfamily of component, manufacturer, component code, topographic marker, family/subfamily hardware box, and datasheet reference
Analysis	Failure: confirmed (yes/no?), failure mode, and failure signature Tests performed Failure mechanisms: common, specific, non-identified/other, and contributing mechanisms Cause: root cause and commonality across the model/series
Corrective actions	Description of corrective actions