



A LANDSCAPE FOR THE DEVELOPMENT OF DEPENDABLE MACHINES

Authored by

IEEE P2851 Working Group

TRADEMARKS AND DISCLAIMERS

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

The ideas and proposals in this specification are the respective author's views and do not represent the views of the affiliated organization.

ACKNOWLEDGEMENTS

Special thanks are given to all the members of the IEEE P2851 WG.

The Institute of Electrical and Electronics Engineers, Inc. 3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2021 by The Institute of Electrical and Electronics Engineers, Inc.

All rights reserved. 30 June 2021. Printed in the United States of America.

PDF: STDVA24820 978-1-5044-7770-3

IEEE is a registered trademark in the U. S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated. All other trademarks are the property of the respective trademark owners.

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/p9-26.html>.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

Find IEEE standards and standards-related product listings at: <http://standards.ieee.org>.

NOTICE AND DISCLAIMER OF LIABILITY CONCERNING THE USE OF IEEE SA DOCUMENTS

This IEEE Standards Association (“IEEE SA”) publication (“Work”) is not a consensus standard document. Specifically, this document is NOT AN IEEE STANDARD. Information contained in this Work has been created by, or obtained from, sources believed to be reliable, and reviewed by members of the activity that produced this Work. IEEE and the IEEE P2851 WG expressly disclaim all warranties (express, implied, and statutory) related to this Work, including, but not limited to, the warranties of: merchantability; fitness for a particular purpose; non-infringement; quality, accuracy, effectiveness, currency, or completeness of the Work or content within the Work. In addition, IEEE and the IEEE P2851 WG disclaim any and all conditions relating to results and workmanlike effort. This document is supplied “AS IS” and “WITH ALL FAULTS.”

Although the IEEE P2851 WG members who have created this Work believe that the information and guidance given in this Work serve as an enhancement to users, all persons must rely upon their own skill and judgment when making use of it. IN NO EVENT SHALL IEEE-SA OR ICAP MEMBERS BE LIABLE FOR ANY ERRORS OR OMISSIONS OR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS WORK, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Further, information contained in this Work may be protected by intellectual property rights held by third parties or organizations, and the use of this information may require the user to negotiate with any such rights holders in order to legally acquire the rights to do so, and such rights holders may refuse to grant such rights. Attention is also called to the possibility that implementation of any or all of this Work may require use of subject matter covered by patent rights. By publication of this Work, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. The IEEE is not responsible for identifying patent rights for which a license may be required, or for conducting inquiries into the legal validity or scope of patents claims. Users are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. No commitment to grant licenses under patent rights on a reasonable or non-discriminatory basis has been sought or received from any rights holder.

This Work is published with the understanding that IEEE and the IEEE P2851 WG members are supplying information through this Work, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. IEEE is not responsible for the statements and opinions advanced in this Work.

TABLE OF CONTENTS

ABSTRACT	5
1. ABOUT IEEE P2851	6
2. THE LIFECYCLE APPROACH OF P2851	7
2.1. LIFECYCLE BACKGROUND	7
2.2. THE DEPENDABILITY LIFECYCLE	11
3. THE LANDSCAPE NEEDS	13
4. CONCLUSIONS.....	17
5. REFERENCES.....	19
APPENDIX A	19

A LANDSCAPE FOR THE DEVELOPMENT OF DEPENDABLE MACHINES

ABSTRACT

This white paper, developed in the framework of the IEEE P2851 standardization project, describes a landscape of activities to be performed to address the dependability of intelligent machines during their development and usage in the field.

In this context, the landscape is intended as the definition of the entirety of the activities that are executed within the autonomous machine dependability lifecycle. Dependability here is intended as the property of an autonomous machine to perform reliably, safely, securely, in a time-deterministic manner, etc.

It should be noted that despite autonomous machines introducing specific dependability challenges, the scope of this document is not limited to full autonomy; rather, it covers the whole spectrum of degrees of automation, from no automation to full automation.

Also worth noting is that nowadays dependable machines are extremely connected. Using automotive as an example, Connected Automated Vehicles (CAV) involves interactions with other vehicles and infrastructure (V2X) and also link with the cloud for functions such as fleet management, teleoperation, maps, Over the Air (OTA) updates, etc. Therefore, the proposed lifecycle is considered end to end, covering the dependability aspects of connectivity as well.

The landscape includes the definition of *needs* for each activity, in terms of methodologies, description languages, data models, and databases that have been identified as necessary or critical to perform those activities.

The goal of this work is to provide structures and directions to allow a seamless exchange of information and interoperability between activities at the same or different level of abstraction.

1. ABOUT IEEE P2851

The IEEE P2851 initiative [1] has been created upon a request of the IEEE Computer Society Special Technical Community (STC) on Reliable, Safe, Secure and Time Deterministic Intelligent Systems (RSSTDIS) [2].

The STC noted that the development of safety and cybersecurity critical systems is rapidly growing due to the expansion of new applications such as automated driving or autonomous mobile robotics.

Standards such as ISO 26262 [4], IEC 61508 [5], and many others define the complete set of activities that need to be performed, requiring companies at different levels of the supply chain to tailor the analysis and verification activities that apply to them and deliver results to other levels of the chain for which other requirements are applicable. EDA vendors have already started to provide tools to automate those activities. However, currently, there are not common methodologies, languages, or formats to provide those results. Even worse, today there is not a cohesive view of all those activities that are executed, often in a disjointed way. As a result of this gap, companies are struggling with many different types of methodologies and description languages and are investing valuable time and effort to reconsolidate, compare, integrate, and combine the data. For this reason, the safety critical community is urgently asking for a solution to accelerate the safety engineering process while reducing risks and costs.

In that context, the IEEE P2851 standardization initiative has been started to define and deliver a comprehensive view of all those activities. One objective is to ultimately define a format for exchange/interoperability for analysis and verification activities (including requirements, safety cases, etc.) and facilitate companies in delivering results in a consistent way. Additionally, the work products of this standardization initiative will enable interoperability between tools.

The initial scope of IEEE P2851 was identified by the RSSTDIS STC as related to Functional Safety at IP and SoC level. However, during the IEEE P2851 kickoff, it was already identified that the scope had to be expanded to systems and items, including software, as well as expanded to cybersecurity and the other properties of dependable systems as in the scope of the RSSTDIS STC: reliability, safety, security, time determinism, and others. Therefore, the end goal of IEEE P2851 is to cover the full spectrum of dependability topics.

2. THE LIFECYCLE APPROACH OF P2851

Several international and industry standards (e.g., ISO 26262 [4], IEC 61508 [5]) propose various details of development lifecycles. These details vary in scope, format, terminology, and consistency of language. This lack of uniformity already presents industry challenges to developments that need to address multiple standards. Given that IEEE P2851 spans several industry domains, it is essential to define the lifecycle in a manner that provides a common language and understanding to enable translation with lifecycles defined by other standards.

2.1. LIFECYCLE BACKGROUND

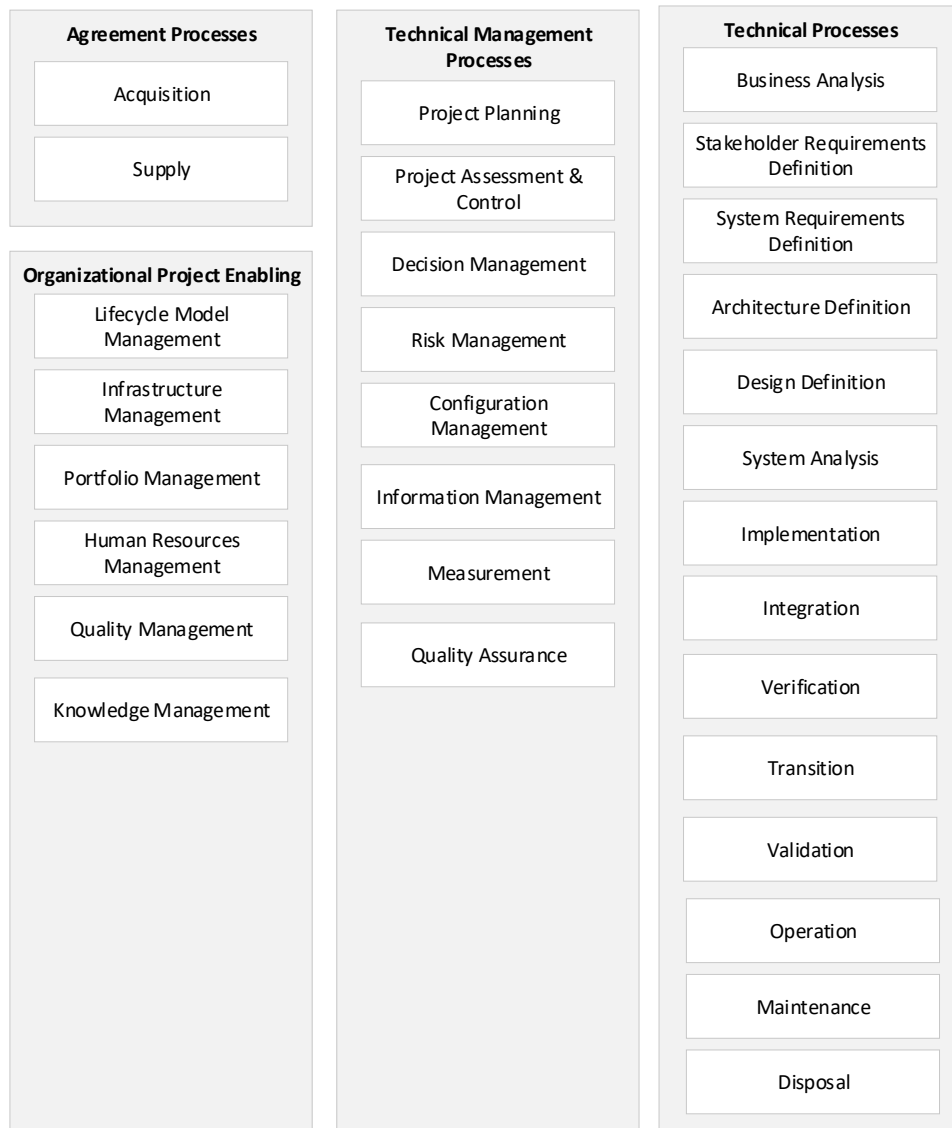
ISO/IEC/IEEE 15288 [6] and ISO/IEC/IEEE 24748 [7] were developed to promote uniformity in lifecycle models and hence are leveraged by IEEE P2851. The following paragraphs provide a brief overview of these standards.

ISO/IEC/IEEE 15288 defines a process as a set of interrelated activities that transform inputs into outputs. A process is described in terms of five characteristics: title, purpose, outcome, activities, and tasks necessary to execute the transformation. This level of description reduces the potential for misinterpretation and improves the ability to translate with other standards. ISO/IEC/IEEE 15288 goes on to define a complete set of processes necessary to enable the development of a product from conception to decommissioning. It groups these processes into the following four categories:

- **Agreement processes:** Concerned with supply and acquisition between organizations and within organizations.
- **Technical processes:** Concerned with technical actions that transform the product from one form to another form.
- **Technical management processes:** Concerned with managing resources and assets to enable the execution of the technical processes in a manner that meets what has been agreed via the agreement processes.
- **Organizational project enabling processes:** Concerned with establishing the environment in which development projects are conducted.

FIGURE 1 summarizes these processes. The intent is not to say that this is the only set of processes by which a product may be developed. Indeed, ISO/IEC/IEEE 15288 allows for the tailoring of processes and acknowledges that an organization may identify additional useful processes.

FIGURE 1 — ISO/IEC/IEEE 15288 System Lifecycle processes



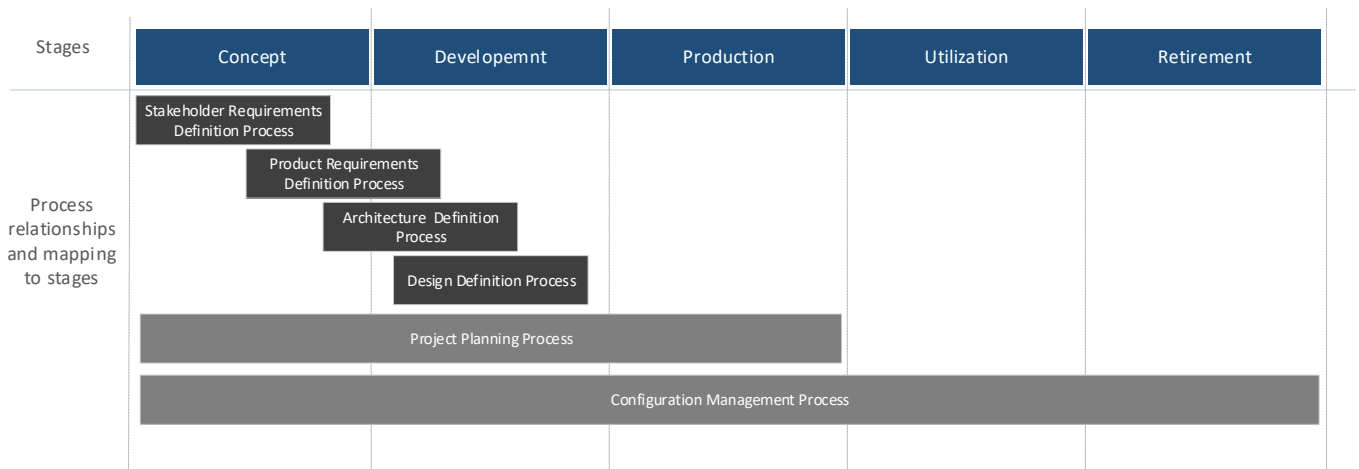
ISO/IEC/IEEE 24748 provides guidelines on how ISO/IEC/IEEE 15288 processes may be arranged to enable product development. A lifecycle model is expressed in terms of the specific processes employed and the relationships and interdependencies between them. It defines a phase to represent a major lifecycle period that relates to the state of the products description or realization. Entry and exit criteria may be defined for each stage, enabling support for strong project control and monitoring. It presents a representative lifecycle model consisting of the following six stages:

- **Concept:** Initial definition of needs and requirements. Exploration of alternative options and proposal of viable solutions.
- **Development:** Refine requirements, create solution description, build system, verify, and validate the system.

- **Production:** Produce system, inspect, and test.
- **Utilization:** Operate system.
- **Support:** Provide sustained system capabilities.
- **Retirement:** Store, archive, or dispose of a system.

FIGURE 2 presents a highly simplified view of a potential mapping for a subset of processes into stages.

FIGURE 2 — Potential mapping of processes to stages

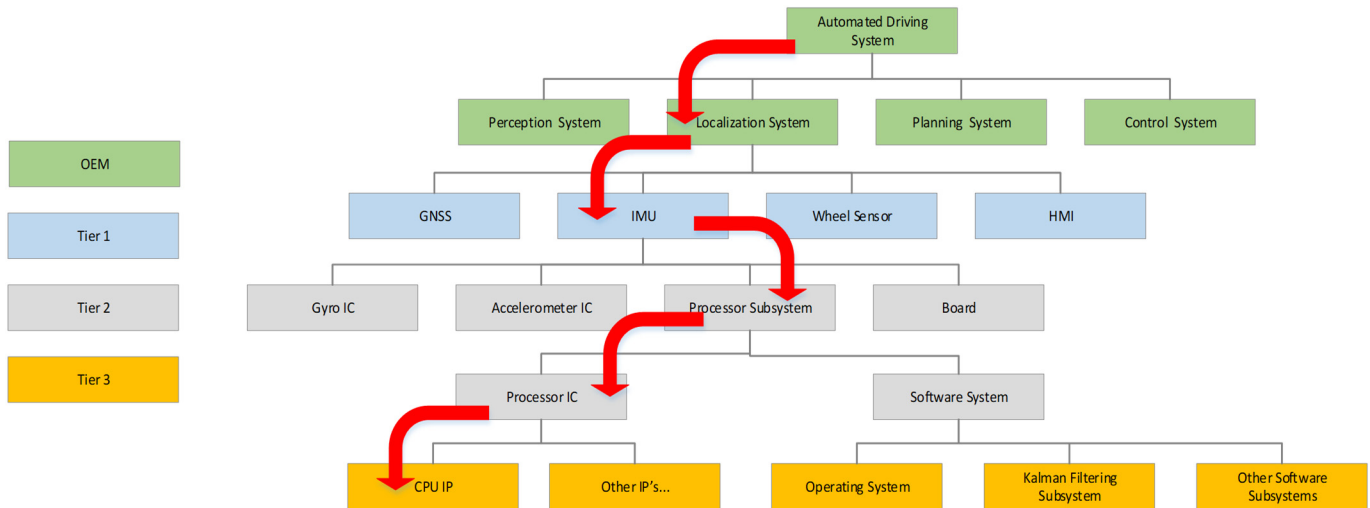


The intent is not to say that this is the only process to stage mapping that could be assigned to development.

Both ISO/IEC/IEEE 15288 and ISO/IEC/IEEE 24748 highlight the importance of the iterative and recursive nature of processes in product development.

Recursive application of process, i.e., the same process applied at different levels of granularity, is expected, as requirements at one level create requirements on lower levels. FIGURE 3 provides an automotive centric example of this recursive nature, taking an automated driving system as an example. The red arrows describe the top-down break down of requirements from top level (item) to the bottom (semiconductor IP level). The automated driving subsystem will place requirements on the localization system. The localization system in turn comprises various subsystems including a Global Navigation Satellite System (GNSS) and an Inertial Measurement Unit (IMU), each of which inherits requirements. The IMU in turn is comprised of multiple integrated circuit and software components all of which inherit their own requirements.

FIGURE 3 — Example of recursive application of processes

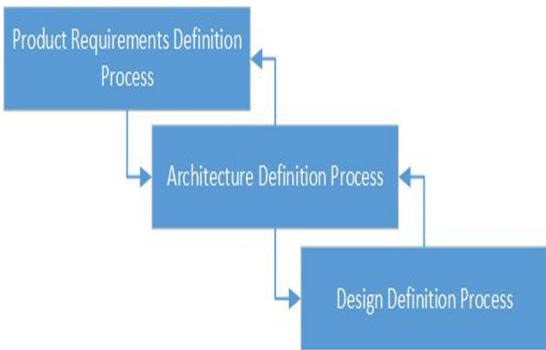


It should be recognized that each of these developments will go through their own development lifecycle and must interface at the appropriate points with the needed levels of granularity.

Iterative application of process, i.e., a process repeated on the same level of product granularity, is not only appropriate but also expected, to account for evolving knowledge and trade-offs. FIGURE 4 presents a simplified view of the iterative relationship between the requirements, architecture, and design processes. The intent is to highlight that some requirements cannot be derived until some portion of the architecture or design evolves. This iterative nature applies to all processes, not only the requirements process.

ISO/IEC/IEEE 24748 further introduces the notion of a development strategy that prescribes certain processes and activities that may be performed sequentially, repeated and/or combined. The lifecycle model and processes are mapped to the desired development strategy. Examples of development strategies include Waterfall, V-model, and Agile developments.

FIGURE 4 — Iterative relationship between requirements, architecture, and design processes



2.2. THE DEPENDABILITY LIFECYCLE

The IEEE P2851 landscape finds it useful to introduce the notion of a Product Dependability Lifecycle (PDL). The term *dependability* has been selected to cover a broad spectrum of properties, including Functional Safety, which is the safety of the intended functionality, cybersecurity, reliability, availability, maintainability, and timing guaranteed.

It is important to recognize that these properties are nothing more than requirements for product development. These requirements are uncovered during the requirements definitions phases of the product development. In an IEEE P2851 PDL, requirements can serve, for example, the purposes of:

- **Functional:** The functions supported by the product. To evaluate the performance of the prescribed functions, the function definition shall be atomic in nature.
- **Performance:** The performance targets necessary to support functional requirements. This would include any timing behavior and **availability** requirements. In an integrated circuit context such requirements could include performance characteristics as diverse as CPU clock frequency, channel phase mismatch, or signal-to-noise targets.
- **Reliability:** Including mission profile and product lifetime. Many other reliability requirements are often inherited from auto reliability standards, e.g., AECQ.
- **Safety of the Intended Functionality (SOTIF):** Typically, understood as the nominal performance necessary to avoid hazards. This may be equal to or a subset of the Performance requirements.
- **Functional Safety:** Requirements related to the avoidance of hazards due to malfunctions. These may introduce their own real-time requirements, for example, the need to execute a safety mechanism within certain timing guarantees.
- **Cybersecurity:** Requirements related to the avoidance of hazards due to deliberate attacks.

It is important to recognize that functions share resources (such as memory space, use of sub-functions). Hence, it is critical that a holistic solution must be arrived at, which always fulfills all requirements. This can be achieved by a combination of smart refinement of requirements, appropriate architecture choice and efficient implementation of the functions. This is a classical system engineering challenge. For example, consider a RADAR transceiver integrated circuit. An on-chip safety mechanism to provide coverage for random faults on the RX and TX paths must share IC resources with the main radar functionality. An architecture and implementation must be found that fulfills the combined performance requirements.

FIGURE 5 presents a simplified view of the need to ensure requirements consistency within an iteration. The iteration could require jumping back to any higher level of the flow as needed. The staggering of Functional, Performance, Reliability, Safety, and Cybersecurity requirements is to convey that first the function should be understood to an adequate level, which then allows performance to be revealed, which in turn allows the safety requirements to be revealed via the appropriate analysis (FMEDA, FTA, etc.).