# A Maturity Model Guidance Approach for Integration Testing of Avionics Software

Gülsüm Güngör [1,2], Ayça Kolukısa Tarhan [1]

[1] *Hacettepe University, Hacettepe Beytepe Kampüsü, Ankara, 06800, Turkey*
[2] *Turkish Aerospace Industries(TAI), Fethiye Mahallesi Havacılık Bulvarı No:17, Ankara, 06980, Turkey*

**Abstract**

Safety-critical software failures lead to serious results such as loss of live or damage to the environment; therefore, safety-critical software verification requires special attention. Avionics system software is one type of safety-critical software. "DO-178C: Software Considerations in Airborne Systems and Equipment Certification" was released in 2011 by RTCA, Inc., which defines processes for airborne systems software development and verification. On the other hand, there are well-defined guidelines to improve verification and validation processes of software system development, specifically for software testing. TMMI (Test Maturity Model Integration) model is produced by TMMI Foundation as a guidance for organizations to improve their test processes and product quality. Avionics system software has own safety-related software characteristics, and TMMI does not specifically address software testing practices of these characteristics. To fill this gap, we first identify avionics software characteristics from DO-178C handbook as the base for software testing, and then propose a domain specific guidance document that employs TMMI practices as complementary to DO-178C activities. The document is aimed to help test organizations in improving test processes by focusing on airborne software characteristics. The proposed approach is targeted for integration testing of avionics software, since this level of testing is very critical for defect prevention in the safety-critical domain.

**Keywords**
Safety-critical, avionics software, integration testing, DO-178C, TMMI, maturity model, test maturity

## 1. Introduction

Safety-critical system failures lead to serious results such as loss of lives or damage to the environment. The software used in avionics systems is classified as safety-critical software in which emerging errors can cause serious consequences. Verification of avionics software is crucial to prevent these undesired results. The first guide to standardize avionics software development was published in 1981 with the name "DO-178: Software Considerations in Airborne Systems and Equipment Certification" [1]. In 2011, DO-178C version [2] was released, which addresses software verification processes with different levels of testing (i.e., requirement-based low-level testing, integration testing, and hardware-software integration testing).

DO-178C defines integration testing as it ensures that software components interact correctly and satisfies software requirements and software architecture [2]. Defects that can be detected only at the level of integration testing are critical to avoid serious consequences in avionics software. Since DO- 178C document heavily focuses on requirement-based integration testing, obeying to DO-178C alone is not sufficient to evaluate and improve integration testing processes.

32nd International Workshop on Software Measurement (IWSM) and the 17th International Conference on Software Process and Product Measurement (MENSURA), September 14–15, 2023, Rome, Italy
gulsumgng@gmail.com (G. Güngör); atarhan@cs.hacettepe.edu.tr (A. K. Tarhan)
0009-0002-3083-7592 (G. Güngör); 0000-0003-1466-9605 (A. K. Tarhan)

On the purpose of testing process and software quality improvements, various models have been developed. Test maturity models such as Test Improvement Model (TIM) [3], Test Process Improvement Model (TPI) [4], Test Maturity Model Integration (TMMI) [5], Unit Test Maturity Model [6] and PTMM [7] are among these models. The mentioned maturity models can be classified in different groups according to their characteristics. The first group can be defined as tester (or person) skills centered maturity models such as PTMM [7]. This type of models focuses on tester skills to improve testing maturity. The second group includes maturity level-based models that each level has its own goals to be achieved to reach a defined maturity level [5]. The third group of maturity models are testing level based models that specifically focus on one testing level (such as unit testing) and offer activities for the concerned level [6]. Another group of test maturity models include continuous models that define key performance areas to determine maturity levels [4]. Also, there are some models applicable on automated testing activities [8]. None of the maturity models mentioned above focuses on integration testing level or avionics software (in safety-critical) testing domain. Similarly, the well- recognized software testing standard ISO/IEC 29119 [9] does not focus on software testing maturity or avionics software testing in particular. TMMI doesn't particularly focus on any domain, but it is a well-designed common model. Furthermore, the structure of TMMI document is similar to DO-178C handbook. In order to fill this gap, this study aims to offer a guidance document for test organizations who want to apply TMMI practices for their integration testing processes within avionics software development.

## 2. Method

The DO-178C handbook defines software development life-cycle processes starting from software planning [2]. In the study [10] entitled "Evaluation of accomplishment of DO-178C objectives by CMMI-DEV 1.3", intersection of CMMI-DEV (Capability Maturity Model Integration for Development) [10] practices and DO-178C activities are defined. Some of the CMMI-DEV practices are matched with the DO-178C activities [10]. It is concluded that CMMI-DEV is not sufficient to cover all the software development activities mentioned in DO-178C and most of the DO-178C verification activities are out of CMMI-DEV's scope [10]. On the other hand, verification and testing activities are in the scope of TMMI since the testing terminology used in TMMI has been derived from ISTQB (International Software Qualifications Board) Standard Glossary of terms used in Software Testing [5]. However, TMMI is a generic testing maturity model and not specific to avionics domain.

In this study, efficient application of TMMI model is aimed for improving avionics software verification process, specifically integration testing process. The DO-178C handbook and the TMMI model are analyzed to understand the necessity for a maturity model that is specific to safety-critical software integration testing. The TMMI model, which can be used complementary to CMMI [5], is found to be convenient to match its processes and practices with the verification activities defined in the DO-178C handbook. Also, TMMI is one of the level-based models that is applicable for all software testing levels, including integration testing, and it covers both manual regression tests and automated tests [5].

In this context, as the first step, processes in DO-178C are inspected and the activities mentioned in DO-178C processes are mapped with the TMMI (Release 1.3) practices, in order to understand the similarity between the two as specific to verification and software testing. After this, DO-178C avionics software characteristics that are not specifically mentioned in the TMMI model identified. As the second step, considering these domain based characteristics, a guidance document for test organizations is proposed to effectively apply TMMI model and improve integration test processes of avionics software. For example, TMMI Level-2 refers to specific goals such as establishing test policy (addressing test goal definition practice) and test strategy [5]. Furthermore, in this level, TMMI practices that aim to achieve these goals refer to determining business objectives, business needs and project needs [5]. Based on the findings of the mapping process, characteristics and needs of DO-178C software integration testing are defined within this guidance document and can be utilized to determine business objectives, business type, or project needs mentioned in TMMI practices.

## 3. Comparison and Results

### 3.1. Comparison between DO-178C and TMMI

Software development life-cycle processes are covered in subtitles of the DO-178C handbook as listed below.

1. Software planning process,
2. Software development process,
3. Software verification process,
4. Software configuration management process,
5. Software quality assurance process,
6. Certification liaison process.

The DO-178C handbook summarizes these software development life-cycle processes within tables in Annex-A [2]. Each process involves objectives and related activities to reach the defined objectives. That is, the tables in Annex-A map activities and objectives for each process [2].

The TMMI model, on the other hand, offers process areas together with their goals and practices to achieve these goals, for each TMMI level [5]. The TMMI model contains five maturity levels, and each level has its own specific practices [5]. Besides, TMMI defines generic practices that are common for all process areas [5].

In the first step of this study, DO-178C process areas are analyzed and each activity mentioned in DO-178C sections are compared with TMMI (Release 1.3) practices, in order to understand the relation between DO-178C verification activities and TMMI test process maturity practices. Each DO-178C activity is compared with the TMMI's specific practices at all maturity levels and with the generic practices. Figure-1 shows the mapping schema of each DO-178C activities to TMMI practices. The analyzed DO-178C activities are grouped as "Covered", "Partially Covered" and "Not Covered" according to the comparison results with the TMMI practices. DO-178 activity that is common for at least one TMMI practice is classified as "Covered". DO-178C activity having scope that is partially matched with any TMMI practice is classified as "Partially Covered". If there is no relevant TMMI practice for the analyzed DO-178C activity, that activity is classified as "Not Covered".

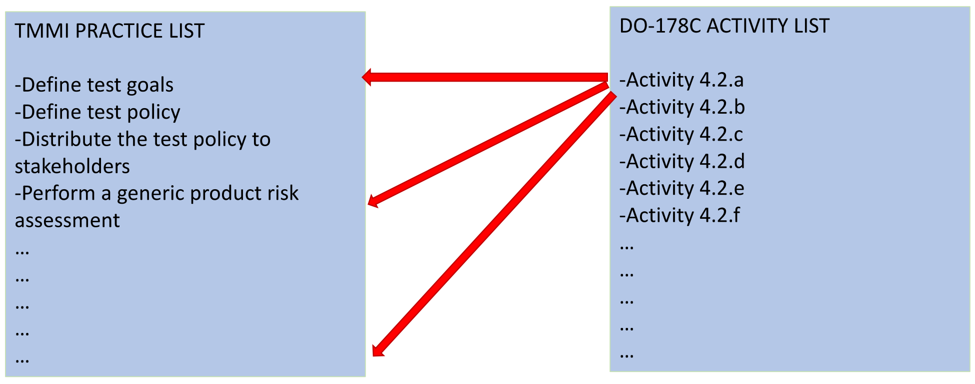


**Figure 1**: Comparison of TMMI practices and DO-178C activities

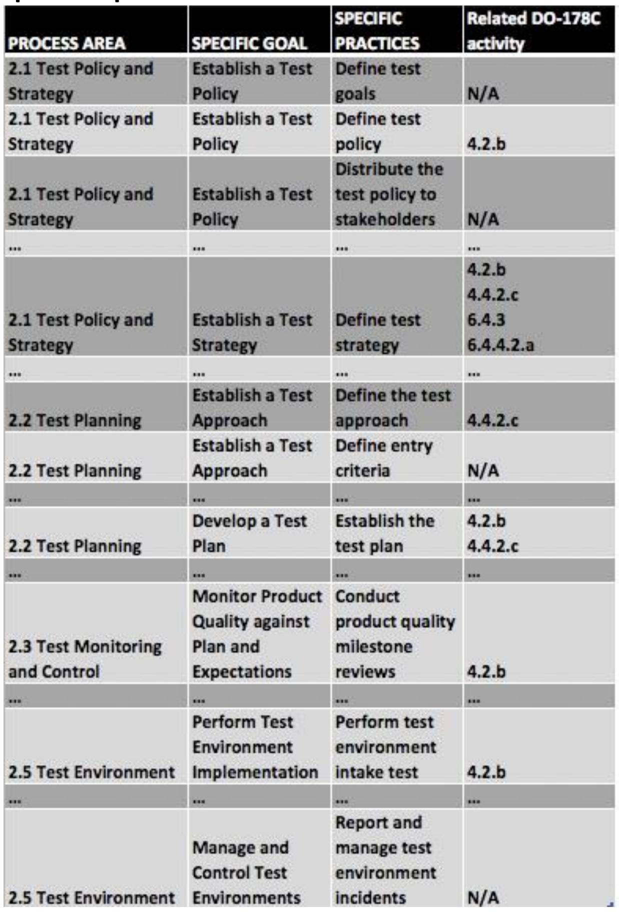
DO-178C activities are analyzed respectively, starting from the first process defined in DO-178C Section-4: Software Planning Process. Table 1 shows a snapshot from the comparison between the activities of software planning process of DO-178C and the TMMI process area practices.

Since TMMI focuses on testing and test planning process areas, they are related only with test planning activities within Software Planning Process of DO-178C. Accordingly, the DO-178C software planning activities are classified as “Covered”, “Partially Covered” and “Not Covered” as shown in Table 1. The DO-178C software planning activities, which are in the scope of “Test Planning” process area of TMMI, are classified as “Covered”, and it has been observed that the number of activities in “Covered” and “Partially Covered” groups corresponds to only half of the practices in this process area. The complete mapping between the TMMI process area practices and the DO-178C process activities can be reached from [11].

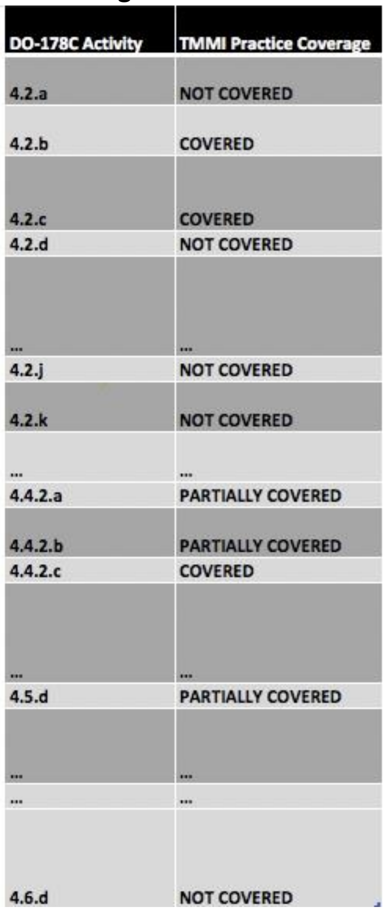
For each software development life-cycle process defined in DO-178C sections, a new table (similar to the one in Table 1) is created per section (or subsection) considering the structure of Annex-A. It should be reminded that the tables in Annex-A summarize the activities [2] of the processes covered in DO-178C. The handbook’s sections of processes from Section-4 to Section-9 include “Software Planning Process”, “Software Development Process”, “Software Verification Process”, “Software Configuration Management Process”, “Software Quality Assurance Process” and “Certification Liaison Process”, respectively [2].

Section-5 in DO-178C defines software development process, and activities of this process are related with software requirements, software design, coding and integration processes but not testing [2]. According to the results of the mapping given in [11], 5.71% of the activities are considered as “Partially Covered” whereas rest of the software development process activities are not covered by TMMI process area practices.

**Table 1**
**An Example Comparison of TMMI Practices and DO-178C “Software Planning Process” Activities.**



| PROCESS AREA | SPECIFIC GOAL | SPECIFIC PRACTICES | Related DO-178C activity |
|---|---|---|---|
| 2.1 Test Policy and Strategy | Establish a Test Policy | Define test goals | N/A |
| 2.1 Test Policy and Strategy | Establish a Test Policy | Define test policy | 4.2.b |
| 2.1 Test Policy and Strategy | Establish a Test Policy | Distribute the test policy to stakeholders | N/A |
| ... | ... | ... | ... |
| 2.1 Test Policy and Strategy | Establish a Test Strategy | Define test strategy | 4.2.b<br>4.4.2.c<br>6.4.3<br>6.4.4.2.a |
| ... | ... | ... | ... |
| 2.2 Test Planning | Establish a Test Approach | Define the test approach | 4.4.2.c |
| 2.2 Test Planning | Establish a Test Approach | Define entry criteria | N/A |
| ... | ... | ... | ... |
| 2.2 Test Planning | Develop a Test Plan | Establish the test plan | 4.2.b<br>4.4.2.c |
| ... | ... | ... | ... |
| 2.3 Test Monitoring and Control | Monitor Product Quality against Plan and Expectations | Conduct product quality milestone reviews | 4.2.b |
| ... | ... | ... | ... |
| 2.5 Test Environment | Perform Test Environment Implementation | Perform test environment intake test | 4.2.b |
| ... | ... | ... | ... |
| 2.5 Test Environment | Manage and Control Test Environments | Report and manage test environment incidents | N/A |



| DO-178C Activity | TMMI Practice Coverage |
|---|---|
| 4.2.a | NOT COVERED |
| 4.2.b | COVERED |
| 4.2.c | COVERED |
| 4.2.d | NOT COVERED |
| ... | ... |
| 4.2.j | NOT COVERED |
| 4.2.k | NOT COVERED |
| ... | ... |
| 4.4.2.a | PARTIALLY COVERED |
| 4.4.2.b | PARTIALLY COVERED |
| 4.4.2.c | COVERED |
| ... | ... |
| 4.5.d | PARTIALLY COVERED |
| ... | ... |
| ... | ... |
| 4.6.d | NOT COVERED |

.

Software verification process in Section-6 is addressed by the following five tables in DO-178C Annex-A [2]: 1) Table A-3: Verification of Outputs of Software Requirements Process, 2) Table A-4: Verification of Outputs of Software Design Process, 3) Table A-5: Verification of Outputs of Software Coding and Integration Processes, 4) Table A-6: Testing of Outputs of Integration Process, and 5) Table A-7: Verification of Verification Process Results. Each table corresponds to sub-sections of Section-6 for software verification process. Some emerging needs and challenges for the practices are observed during the comparison of this process, as summarized below:

- DO-178C refers to separate sub-sections for low-level and high-level requirements analyses [2]; however, the TMMI model does not particularly mention about testing practices of low-level or high-level requirements since it can be applied to improve the practices at any testing level. Accordingly, the activities in Section 6.3.1: Reviews and Analyses of High-Level Requirements and 6.3.2: Reviews and Analyses of Low-Level Requirements do not match with the TMMI process area practices specifically.
- DO-178C Section 6.4 defines software testing process, and different levels of testing activities, including the ones for integration testing, are described in this process [2]. Therefore, the 65.21% of the activities mentioned in this process with the TMMI practices is identified as quite high.
- DO-178C Section 6.5 addresses software verification process traceability [2], and the activities are considered as "Covered" and "Partially Covered" by the TMMI practices for this subsection since DO-178C defines three bi-directional traceability analyses which are between software requirements and test cases, test cases and test procedures, test procedures and test results.
- DO-178C has avionics software development characteristics and "parameter data item", which is a feature to enable changing behavior of software without modifying its code, is one of them as mentioned in Section 6.6 [2]. Since TMMI is not a domain specific test maturity model, it does not offer any verification or test maturity practice for parameter data items.

Section 7 in DO-178C defines software configuration management process [2] and TMMI considers configuration management in its general practices by referring to CMMI's configuration management process [5]. The study [10] provides coverage rate as 48% between CMM-Dev and DO-178C for configuration management activities. In this current study, activities of this process are considered by TMMI practices and the practices labeled both "Covered" and Not Covered" are found. Software quality assurance process is defined in DO-178C Section 8 [2] and its activities are also considered in the groups of "Covered" or "Not Covered". Finally, DO-178C Section 9: Certification Liaison Process [2] is out of the TMMI's scope, and therefore, the activities belonging to this process are "Not Covered" by any TMMI practice.

As a result, it has been observed that the TMMI practices are not sufficient alone to accomplish DO- 178C verification activities. DO-178C has avionics software development characteristics and definitions that are not in the scope of TMMI. DO-178C defines avionics software specific items and concepts such as verification of parameter data item, user modifiable software, deactivated code, multi- version dissimilar software verification, COTS software, option selectable software, and field-loadable software [2] that are not discussed in TMMI. Therefore, some of the process activities in DO-178C (regarding verification and testing of avionics software) do not match with the TMMI practices. For example; low-level requirements, high-level requirements, and their relation to system requirements are mentioned in detail within DO-178C objectives, but TMMI process area goals do not refer to this structure which is specific to avionics domain. In addition, some of the change related activities (software change, requirement change, new compiler usage, different loader version, change of application or development environment, etc.) and re-execution needs of tests are defined in DO-178C within safety-critical aspects [2]. Even though TMMI offers change related practices [5], scope of the change should be revised by considering DO-178C safety-critical avionics software development. Moreover, the DO-178C handbook includes a subsection for “Robustness Test Cases” that shows software behavior in abnormal conditions [2]. It is critical to avoid undesired results, but it is not particularly discussed in TMMI practices.

The shortages mentioned above should not be considered as weaknesses of the TMMI model since it is a generic maturity model that offers many practices to improve testing processes and product quality. Rather, the shortages indicate the need for a testing maturity model specific to avionics domain. Finally, on the opposite side of the mapping, some TMMI process areas such as test training programs, incident management and advanced reviews [5] are not discussed in DO-178C processes in detail. Integration testing is one of the critical test levels in the scope of high-level requirements-based testing in DO-178C to avoid undesired results of safety-critical avionics software. It is observed that TMMI process area practices can enrich the activities for integration testing level defined in DO-178C. Therefore, the results of bi-directional comparison between DO-178C activities and TMMI practices have shown that the mutual consideration of these two resources for developing a domain specific maturity model guidance approach for integration testing of avionics software is prominent.

## 3.2. Guidance Document for TMMI Applications on Avionics Software Integration Testing

In the previous section, the contents of the TMMI model and the DO-178C handbook are compared to understand the requirements of avionics software testing maturity concept. Avionics software characteristics need more specific testing practices to comprise avionics software item verification activities. As the next step, a guidance document that employs TMMI practices as complementary to DO-178C activities is developed to effectively improve domain specific software testing processes, more specifically integration testing activities, within avionics software development.

After determining domain specific characteristics from DO-178C handbook, each TMMI practice is reviewed and relevant DO-178C handbook section references are provided for practices to implement them considering domain specific characteristics. The reference from a TMMI practice to DO-178C section is called link. To apply a TMMI practice, links would be helpful to describe and implement given practice within safety-critical avionics software characteristics. Table 2 shows a link example from guidance document for the TMMI practice called “Test Policy and Strategy” [2,5].

**Table 2**
**An Example from Guidance Document for “Perform a generic product risk assessment” practice**

| TMMI PROCESS AREA | TMMI GOAL | TMMI PRACTICE | LINK TO RELEVANT DO-178C SECTIONS |
|---|---|---|---|
| 2.1 Test Policy and Strategy | Establish a Test Strategy | Perform a generic product risk assessment | 1) Refer to DO-178C section 2.3.2 - failure condition categorization<br>2) Refer to DO-178C section 2.3.3- software level definition |

Links contain DO-178C section references and each referred section defines activities or definitions related to airborne software.

TMMI document defines specific goals and practices starting from TMMI Level-2. In this level, TMMI refers to the specific goal of “Establish a Test Policy” that contains specific practices called

“Define test goals” and “Define test policy” [5]. These practices refer to determining business objectives, business needs and project needs [5]. Also, other TMMI Level-2 process areas such as “Test Planning” and the processes at further maturity levels (from level-3 to level-5) describe practices addressing test goals and strategy that are defined in level-2 practices previously [5]. The business objectives and needs, in other words test policy and strategy that are defined based on domain specific characteristics, can help the implementation of the rest of the TMMI processes considering domain specific characteristics. Therefore, a guidance document that offers links to airborne specific items of DO-178C sections is proposed to employ TMMI practices for improving integration testing of avionics software focusing on domain specific needs.

The list of items mentioned in the DO-178C sections 6.1, 6.4.3 in the following [2]:

1. Software components must interact correctly with each other.
2. System requirements allocated to software have been developed into high-level requirements that satisfy those system requirements.
3. Software must satisfy high-level requirements and responds as expected.
4. Software is robust to respond correctly when abnormal inputs and conditions occurred.
5. Software components together satisfy the software requirements and software architecture.

These items mentioned above are relevant to TMMI level-2 “Define test goals” practice therefore, a link is defined for this practice that refers DO-178C sections 6.1 and 6.4.3. The items can be used to guide test organizations on the application of “Define test goals” practice of TMMI considering them within the business needs and objectives [5]. “Define test goals” practice refers to a sub-practice called “Define test goals traceable to business needs and objectives” and the verification of guidance document items mentioned above can be example to test goals of avionics software integration tests. TMMI document refers to these test goals in further maturity level practices. Therefore, links offered by this guidance approach for a practice can affect other practices if it contains same terms or it is referred by other practices.

## 4. Challenges

In the first step of this study, a comparison process was implemented. Since the coverage decisions in comparison process are subjective, there is a risk to get exactly the same coverage labels for each practice on every repeat by other practitioners. Nevertheless, even different practitioners execute the same comparison process, still they will see that DO-178C activities are not fully covered by TMMI practices so at the end, different outcome isn’t expected.

## 5. Conclusion and Future Work

This study explains the preliminary steps taken to propose an avionics software testing maturity model in order to improve integration testing processes of projects obeying to DO-178C requirements. In this context, DO-178C is analyzed to understand avionics software verification activities and needs. TMMI is taken as the base maturity model since its “process area & practice” structure is similar to the process structure in DO-78C. In the first step, TMMI practices and DO-178C activities are analyzed for bi-directional mapping with respect to the needs of avionics software testing. Then, a document as complementary to DO-178-C handbook is proposed to help test organizations on applying TMMI to improve their integration testing processes within avionics software development, focusing on the characteristics of airborne software domain as described in DO-178C.

As the next step, empirical validation studies are planned to review the complementary (guidance) document and improve it as necessary. DO-178C characteristics and links are defined in this study as a guidance approach.

First, expert opinions will be taken to finalize the scope and content of the proposed guidance document considering links and directions mentioned in it. Then, case study research method will be used to investigate the applicability and usefulness of the final version of the document. Also, maturity level for the case study will be defined and reported according to this approach.

## 6. References

[1] DO-178: Software Considerations in Airborne Systems and Equipment Certification, Washington, DC: RTCA Inc., 1981.

[2] DO-178C: Software Considerations in Airborne Systems and Equipment Certification, Washington, DC: RTCA Inc., 2011.

[3] Ericson, T., Subotic, A. and Ursing, S. (1997), TIM—a test improvement model. Softw. Test. Verif.

[4] J. Andersin, "TPI -a model for Test Process Improvement," 2004. Accessed: Jun. 01, 2023. [Online]. Available: https://www.cs.helsinki.fi/u/paakki/Andersin.pdf

[5] TMMi Foundation, "Test Maturity Model integration (TMMi®) Guidelines for Test Process Improvement Release 1.3 Produced by the TMMi Foundation." Accessed: Jun. 01, 2023. [Online]. Available: https://www.tmmi.org/tmmi-documents/

[6] D. M. Karr, "The Unit Test Maturity Model" Accessed: Jun. 01, 2023. [Online]. Available: http://davidmichaelkarr.blogspot.com/2013/01/the-unit-test-maturity-model.html

[7] S. Reid, "Personal Test Maturity Matrix", Accessed: Jun. 01, 2023. [Online]. Available: https://www.stureid.info/stuart-reid-software-testing/software-testing-white-papers/personal-test- maturity-matrix/

[8] V. Garousi, M. Felderer and T. Hacaloğlu, "What We Know about Software Test Maturity and Test Process Improvement," in IEEE Software, vol. 35, no. 1, pp. 84-92, January/February 2018, doi: 10.1109/MS.2017.4541043.

[9] ISO/IEC/IEEE 29119-1:2022 Software and systems engineering — Software testing — Part 1: General concepts, ISO/IEC/IEEE 29119-1:2022, 2022. [Online]. Available: https://www.iso.org/standard/81291.html

[10] A. Ferreirós and L. A. V. Dias, "Evaluation of Accomplishment of DO-178C Objectives by CMMI- DEV 1.3," 2015 12th International Conference on Information Technology - New Generations, Las Vegas, NV, USA, 2015, pp. 759-760, doi: 10.1109/ITNG.2015.132.

[11] G. Güngör, TMMI&DO-178C Mapping, Accessed: June 2023.[Online]. Available: https://doi.org/10.5281/zenodo.8002215