

ECSEL Research and Innovation actions (RIA)



AMASS

Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems

Evaluation Framework and Quality Metrics D1.3

Work Package:	WP1: Case studies and benchmarking
Dissemination level:	PU = Public
Status:	Final
Date:	30 September 2017
Responsible partner:	Tomáš Kratochvíla (Honeywell)
Contact information:	Tomas.Kratochvila@honeywell.com
Document reference:	AMASS_D1.3_WP1_HON_V1.0

PROPRIETARY RIGHTS STATEMENT

This document contains information, which is proprietary to the AMASS consortium. Permission to reproduce any content for non-commercial purposes is granted, provided that this document and the AMASS project are credited as source.

This deliverable is part of a project that has received funding from the ECSEL JU under grant agreement No 692474. This Joint Undertaking receives support from the European Union's Horizon 2020 research and innovation programme and from Spain, Czech Republic, Germany, Sweden, Italy, United Kingdom and France.

Contributors

Names	Organisation
Tomáš Kratochvíla, Petr Bauch, Vít Koksa	Honeywell International
Miguel Gómez, Juan Castillo	Thales Alenia Space España
Benito Caracuel, David Pampliega	Schneider Electric
Garazi Juez, Huáscar Espinoza, Alejandra Ruiz	Tecnalia Research & Innovation
Stefano Puri	Intecs
Barbara Gallina, Shankar Iyer	Maelardalen Hoegskola (MDH)
Mathias Killer, Yu Bai, Behrang Monajemi	Berner & Mattner Systemtechnik
Elena Alaña, Javier Herrero	GMV Aerospace and Defense
Thomas Gruber, Zhendong Ma, Christoph Schmittner	Austrian Institute of Technology
Thierry Lecomte, David Déharbe	ClearSy
Jose Luis de la Vara, Eugenio Parra, Jose María Álvarez	Universidad Carlos III de Madrid
Luis Alonso, Borja López	The REUSE Company

Reviewers

Names	Organisation
Erwin Schoitsch (peer review)	Austrian Institute of Technology
Jose Luis de la Vara (peer review)	Universidad Carlos III de Madrid
Cristina Martínez (Quality Manager)	Tecnalia Research & Innovation

TABLE OF CONTENTS

Executive Summary.....	7
1. Introduction	8
1.1 Scope.....	8
1.2 Structure of the Document	9
2. Common Evaluation Foundation	10
2.1 Goal-Question-Metric Approach.....	10
2.2 AMASS Goals and Objectives	11
2.3 Software Related Metrics	22
2.4 Common Evaluation Framework	22
3. AMASS Metrics	23
3.1 Reducing Assurance and Certification/Qualification Effort	23
3.1.1 Automation of Architecture-Driven and Multi-Concern Assurance	23
3.1.2 Identification of the Needs of Architecture-Driven and Multi-Concern Assurance	24
3.1.3 Architecture-Driven and Multi-Concern Assurance Reuse.....	24
3.1.4 Architecture-Driven and Multi-Concern Assurance Risks.....	25
3.1.5 Early Identification Impact.....	25
3.2 Reusing Assurance Results.....	26
3.2.1 Reusing Architecture-Driven Assurance Results.....	26
3.2.2 Reusing Multi-Concern Assurance Results	26
3.2.3 Reuse of Certification and Qualification Results	27
3.2.4 Cross-Domain Reuse of Assurance Results.....	27
3.3 Reducing Assurance and Certification and Qualification Risks.....	28
3.3.1 Reducing Risks by Architecture-Driven Assurance	28
3.3.2 Reducing Risks by Multi-Concern Assurance	29
3.3.3 Reducing Risks by Seamless Interoperability.....	29
3.3.4 Reducing Risks by Cross-Domain Assurance.....	30
3.4 Sustainable Impact by Harmonization and Interoperability	30
3.4.1 Sustainable impact by architecture-driven assurance	31
3.4.2 Sustainable impact by multi-concern assurance	31
3.4.3 Sustainable impact by seamless interoperability	32
3.4.4 Sustainable impact by cross-domain assurance	32
3.4.5 Sustainable impact by AMASS eco-system.....	32
3.4.6 Sustainable impact by AMASS community	33
3.5 Common Evaluation Procedures.....	33
3.6 Common Evaluation Framework	33
4. Technical Solution Metrics, Processes, and Tools	35
4.1 Metrics from WP3 – Architecture Driven Assurance	35
4.2 Metrics from WP4 – Multi-Concern Assurance	35
4.3 Metrics from WP5 – Seamless Interoperability	36
4.4 Metrics from WP6 – Cross/Intra Domain Reuse.....	37
5. Case Study-Specific Metrics, Processes, and Tools.....	39
5.1 Case Study 1	43
5.1.1 Metrics	44
5.1.2 Processes and Tools.....	46
5.2 Case Study 2	48
5.2.1 Metrics	48
5.2.2 Processes and Tools.....	49



5.3	Case Study 3	50
5.3.1	Metrics	50
5.3.2	Processes and Tools	51
5.4	Case Study 4	52
5.4.1	Metrics	52
5.4.2	Processes and Tools	53
5.5	Case Study 5	53
5.5.1	Metrics	53
5.5.2	Processes and Tools	54
5.6	Case Study 6	54
5.6.1	Metrics	55
5.6.2	Processes and Tools	57
5.7	Case Study 7	57
5.7.1	Metrics	58
5.7.2	Processes and Tools	59
5.8	Case Study 8	61
5.8.1	Metrics	61
5.8.2	Processes and Tools	61
5.9	Case Study 9	62
5.9.1	Metrics	62
5.9.2	Processes and Tools	64
5.10	Case Study 10	64
5.10.1	Metrics	64
5.10.2	Processes and Tools	67
5.11	Case Study 11	68
5.11.1	Metrics	68
5.11.2	Processes and Tools	68
6.	Conclusions	70
	Abbreviations and Definitions	71
	References	72
	Appendix A Evaluation – EPF Process Description	73

List of Figures

Figure 1.	Relationship of other deliverables with D1.3 Evaluation Framework and Quality Metrics	8
Figure 2.	GQM Definition Procedures, the diagram copied from [9]	11
Figure 3.	AMASS Goal 1 and Goal 2 mapped to Questions that are mapped to Metrics	13
Figure 4.	AMASS Goal 3 mapped to Questions that are mapped to Metrics	14
Figure 5.	AMASS Goal 4 mapped to Questions that are mapped to Metrics	14
Figure 6.	Evaluation process of CoPQ improvement	58
Figure 7.	Top-level overview of the evaluation process	73
Figure 8.	Activity diagram of the planning process	74
Figure 9.	Description of the task <i>Write general sections of D1.3</i>	75
Figure 10.	Description of the task <i>Select improvement areas</i>	75
Figure 11.	Description of the task <i>Describe Evaluation in EPF Composer</i>	76
Figure 12.	The Definition phase of the evaluation consists of just one major task	77
Figure 13.	The description of the task <i>Write detailed sections of D1.3</i>	78
Figure 14.	Activity diagram for the <i>Data collection</i> phase	79
Figure 15.	The description of the task <i>Measure effort</i>	80
Figure 16.	An example of the table with the recorded effort	81
Figure 17.	The description of the task <i>Measure number of items</i>	82
Figure 18.	The overview of the Interpretation phase	83
Figure 19.	The description of the task <i>Analyse collected data and answer questions</i>	83
Figure 20.	The roles that support the Goal-Question-Metric approach	84
Figure 21.	The roles of the development team	85
Figure 22.	The description of the deliverable <i>D1.3 Evaluation Framework and Quality Metrics</i>	86
Figure 23.	The description of the deliverable <i>D1.7 Case study implementation and benchmarking</i>	86
Figure 24.	The process description model	86
Figure 25.	The measured data artifact	87

List of Tables

Table 1.	Assessment of AMASS goal G1	15
Table 2.	Assessment of AMASS goal G2	17
Table 3.	Assessment of AMASS goal G3	18
Table 4.	Assessment of AMASS goal G4	20
Table 5.	Metrics and their usage in the Case Studies	39
Table 6.	CS1 tools and processes	47
Table 7.	CS3 tools and processes	51
Table 8.	CS5 tools and processes	54
Table 9.	CS6 tools and processes	57
Table 10.	CS7 tools and processes	61
Table 11.	CS8 tools and processes, A	62
Table 12.	CS8 tools and processes, B & C	62
Table 13.	CS10 tools and processes	67
Table 14.	CS11 tools and processes	69

Executive Summary

Given the heterogeneity of the AMASS case studies [2], the task T1.3 (Benchmarking Framework) aims to harmonise the evaluation procedures for AMASS results. The AMASS technology partners will provide the guidelines and a common framework that will be used by case study providers and end users to assess the benefit and limitations of the AMASS solution. In addition, end users will define the metrics for the measurements of the quality of the approach. Some metrics describe the properties of processes, while others are focused on the properties of artefacts.

The process-related metrics quantify e.g. the time consumed by a process, or the number of defects introduced or uncovered by a process. While the measurement of time can be less precise (e.g. due to omissions to start/stop the timewatch) and depend significantly on the measurement approach and in some cases on the willingness/discipline of the engineer, the defect related reports from a tracking system might be less subjective.

The artefact-related metrics will usually be generated by the tool that manages the artefacts. In order to automate the collection of such data, it is desirable to either adjust the metrics to the available functionality of the tools, or it could be necessary to develop a new plug-in of the AMASS tool platform that will automatically report the required metrics.

The minimum information that each case study description will provide to AMASS is:

- Procedures to evaluate the AMASS solution in the context of industrial case studies.
- Validation scenarios and test cases containing a flow description of the current process and the expected AMASS-based enhanced process.
- Metrics to measure improvements achieved by the AMASS results in each validation scenario. This includes metrics prioritization and limitations of measurements in real scenarios.

The final questions and metrics to be used are formulated in this document, D1.3 (Evaluation Framework and Quality Metrics) and will be measured in D1.7 (AMASS Solution Benchmarking) [2].

1. Introduction

With increasing complexity of the safety-critical embedded systems, assurance and certification becomes extremely expensive. AMASS aims at reducing these efforts.

This deliverable defines an evaluation framework that consists of the set of metrics for assessing the achievements of AMASS goals. These goals G1, G2, G3, and G4 aim at reducing the assurance and certification/qualification effort and risks, reusing assurance results, increasing the interoperability of assurance and certification/qualification technologies. These goals are defined below in Section 2.2.

Given the heterogeneity of the AMASS case studies [2], the task T1.3 (Benchmarking Framework) aims to harmonise the evaluation procedures for AMASS results. The AMASS technology partners will provide the guidelines and a common framework that will be used by case study providers and end users to assess the benefit and limitations of the AMASS solution. In addition, end users will define the metrics for the measurements of the quality of the approach.

1.1 Scope

The evaluation framework will start with common metrics that are derived from AMASS goals and objectives. This framework is then extended by technical metrics for each case study and specialized metrics for each technical solution from WP3 (Architecture-driven Assurance), WP4 (Multi-concern Assurance), WP5 (Seamless Interoperability), and WP6 (Cross-Domain and Intra-Domain Reuse). Finally, the specialized case study metrics, processes and tools are described.

Figure 1 provides more details on the relationship of D1.3 with other AMASS deliverables.

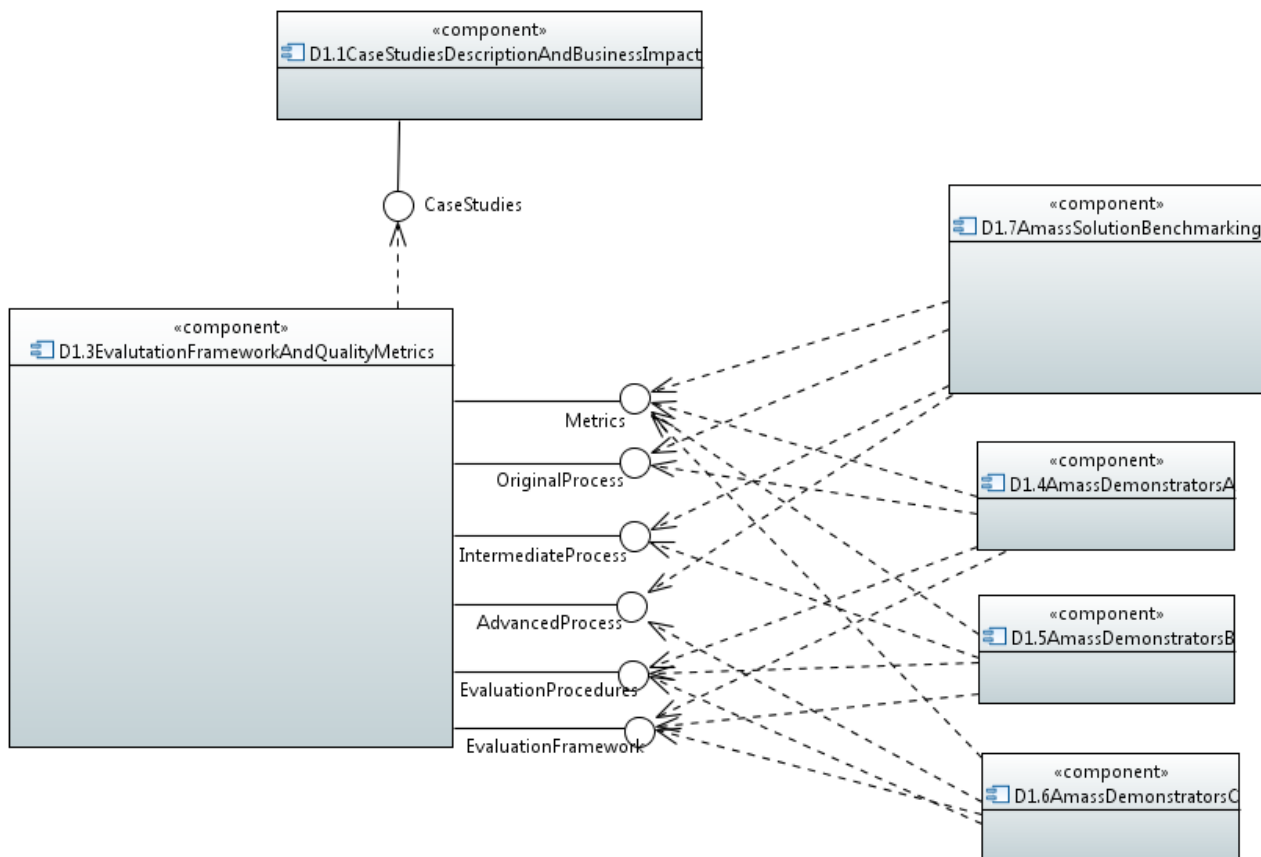


Figure 1. Relationship of other deliverables with D1.3 Evaluation Framework and Quality Metrics

1.2 Structure of the Document

The rest of the deliverable is organised as follows:

- *Chapter 2* introduces the Goal – Question – Metric (GQM) methodology for evaluation of processes and products.
- *Chapter 3* focuses on the definition of the basic/common metrics that are needed for the evaluation of the AMASS platform.
- *Chapter 4* contains the metrics that are important from the perspective of the individual Work Package groups (WP3, WP4, WP5, and WP6).
- *Chapter 5* goes down to the level of individual Case Studies and presents the case study specific metrics and some environment details that are relevant to the evaluation.
- *Appendix A* provides an overview of the evaluation process in the form of a published EPF process.

2. Common Evaluation Foundation

Several case studies shall be performed in order to assess the impact of AMASS and optimize the structure and functionality of the AMASS platform. These case studies are performed by different partners in different application domains and in various case study-specific environments (tools, processes). In order to keep the results of the individual case studies comparable, we need to define common core concepts and general guidelines that will be shared across all the case studies. Such concepts and guidelines are provided in this chapter.

2.1 Goal-Question-Metric Approach

AMASS follows the Goal-Question-Metric (GQM) approach [9] in order to measure the progress of the project and to evaluate the effectiveness of the proposed technologies. This approach has been widely used for product and process assessment, including improvement assessment.

GQM evaluation results in the specification of a measurement system targeting a particular set of issues and a set of rules for the interpretation of the measurement data. The GQM model has three levels [3]:

1. **Conceptual level (Goal).** A goal is defined for an object of measurement, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment. Objects of measurement are:
 - Products: artefacts, deliverables and documents that are produced during system lifecycle; e.g., specifications, designs, programs, and test suites.
 - Processes: software related activities normally associated with time; e.g., specifying, designing, testing, and interviewing.
 - Resources: items used by processes in order to produce their outputs; e.g., personnel, hardware, software, and office space.
2. **Operational level (Question).** A set of questions is used to characterise the way the assessment/achievement of a specific goal is going to be performed based on some characterizing model of quality. Questions try to characterise the object of measurement (product, process, or resource) with respect to a selected quality issue and to determine its quality from the selected viewpoint.
3. **Quantitative level (Metric).** A set of data is associated with every question in order to answer it in a quantitative way. The data can be:
 - Objective, if they depend only on the object that is being measured and not on the viewpoint from which they are taken; e.g., number of versions of a document, staff hours spent on a task, and size of a program.
 - Subjective, if they depend on both the object that is being measured and the viewpoint from which they are taken; e.g., readability of a text and level of user satisfaction.

To define the AMASS GQM measurement program, the GQM definition procedure [9] shown in Figure 2 is followed.

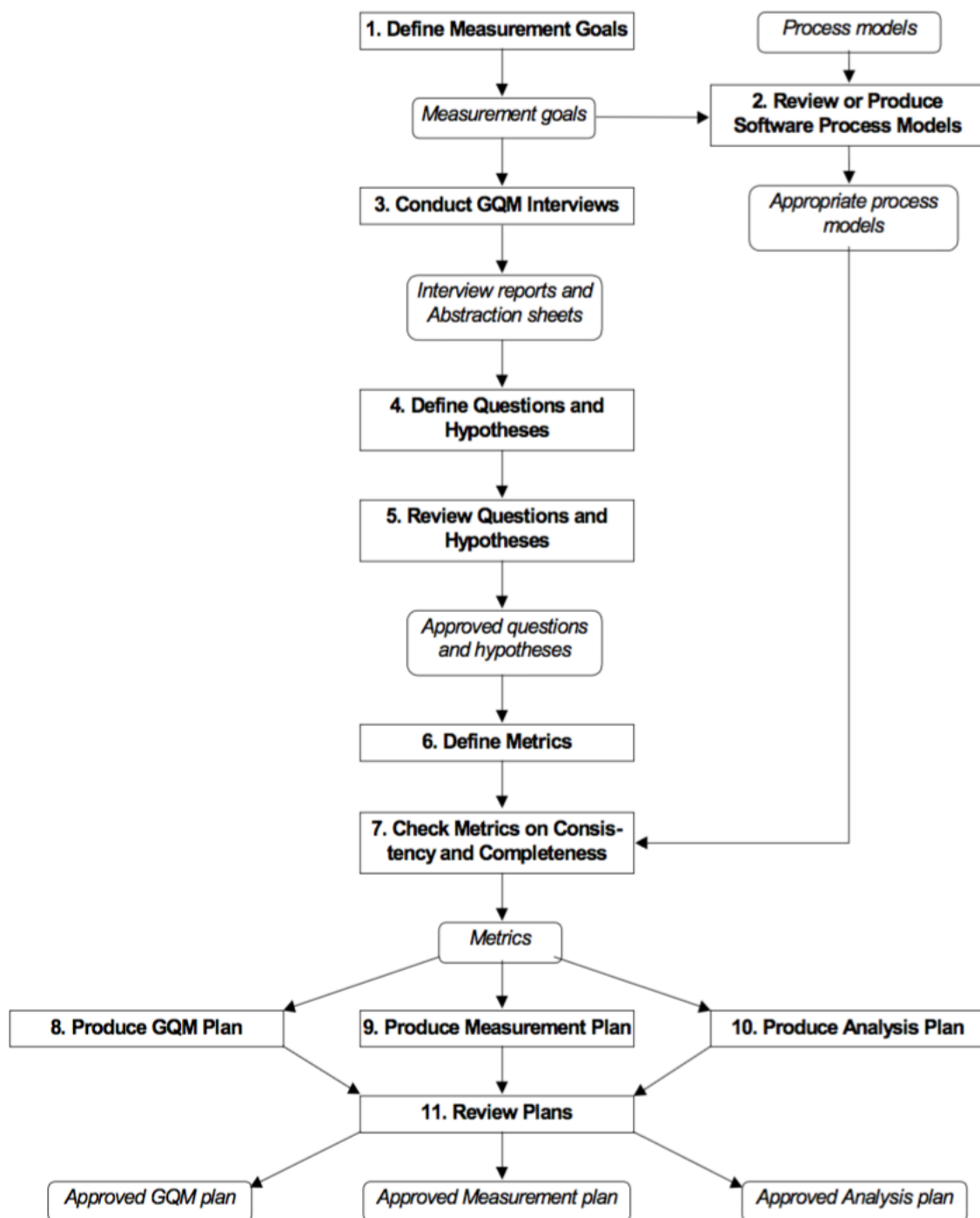


Figure 2. GQM Definition Procedures, the diagram copied from [9]

2.2 AMASS Goals and Objectives

This section lists the goals and objectives as defined in AMASS Description of Work [2].

The **overall goals** of AMASS, to improve the current situation in CPS design technologies, are:

- **G1:** to demonstrate a potential gain for design **efficiency** of complex CPS by reducing their assurance and certification/qualification effort by 50%.
- **G2:** to demonstrate a potential **reuse** of assurance results (qualified or certified before), leading to 40% of cost reductions for component/product (re)certification/qualification activities.

- **G3:** to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification **risks** of new CPS products. The reduction of risks can be “invested” into the risky adoption of new technologies, for which there was no space without the reduction.
- **G4:** to demonstrate a potential sustainable impact in CPS industry by increasing the **harmonization** and **interoperability** of assurance and certification/qualification tool technologies by 60%.

The overall goal achievements will be assessed by comparing estimated end-of-project levels with beginning-of-project levels for the AMASS partners. Metrics behind these numbers are presented in Section 2.4.

The AMASS focus and overall goals will be achieved by means of the following **project objectives**:

- **O1:** define a holistic approach for **architecture-driven assurance** to leverage the reuse opportunities in assurance and certification by directly and explicitly addressing current technologies and HW/SW architectures needs.
- **O2:** define a **multi-concern assurance** approach to ensure not only safety and security, but also other dependability aspects such as availability, robustness and reliability.
- **O3:** consolidate a **cross-domain and intra-domain assurance** reuse approach to improve mutual recognition agreement of compliance approvals and to help assess the return of investment of reuse decisions.
- **O4:** develop a fully-fledged open tool platform that will allow developers and other assurance stakeholders to guarantee **seamless interoperability** of the platform with other tools used in the development of CPSs.
- **O5:** benchmark the tool infrastructure against real industrial cases in relevant environments.
- **O6:** consolidate the AMASS **ecosystem** and **community** for:
 - **O6.a:** adoption of the AMASS conceptual and methodological approach as a **Reference Tool Architecture** for CPS assurance and certification/qualification.
 - **O6.b:** maintenance and further development of the open **AMASS Tool Platform** as a long-term, API-standardized and industry-driven assurance environment.

Indicators of success

We have described the overall AMASS goals and its project objectives above. The overall goals seek to demonstrate measurable gains for AMASS tool platform users, in terms of efficiency, risks, reuse, and harmonization, by comparing estimated levels at the end of the project with the levels at the beginning. The numbers used in the overall goals are derived from internal discussions among the partners and based on estimates coming from practical experience in the field and from the state of the art (e.g., results from SafeCer [23]). There are no scientific arguments to back up those numbers. There are some estimation measurements about similar goals in the OPENCROSS project [22] and they are sound in terms of order of magnitude of the expected benefits. In OPENCROSS, the estimation measurements were provided by the industrial partners and validated by the external advisory board.

The quantitative targets of the goals have also been calculated according to the specific metrics that we provide in the tables below. The metrics and their values correspond to an initial, rough analysis of the current state of the practice and the envisioned improvements, agreed upon by the AMASS consortium. Since we will not know for sure whether the objectives have been achieved and how until the project ends, the AMASS goals defined elsewhere contain the phrase “potential”, and not just reduction or increase. The term “potential” needed for the general description of the goals would turn into the term “real” in the context of concrete projects, measurements, and evaluations.

AMASS will follow the Goal-Question-Metric approach in order to measure the progress of the project and to evaluate the effectiveness of the proposed technologies. The tables Table 1, Table 2, Table 3, and Table 4 show which project objectives and which STO contribute to each overall goal. The tables also provide a set of aspects to address for evaluating each goal. The questions are used to characterise the way in which a

specific overall goal can be assessed, and correspond to issues that are expected to have major impacts on the achievement of the goals. The tables also present a set of metrics that will be used to assess the achievement of the overall goals. These metrics will be refined during AMASS execution according to the needs and characteristics of each case study. Some metrics are case study-independent and can be measured by comparing AMASS results with those of related projects.

The final questions and metrics to be used will be formulated in revisions of D1.3 and will be measured in D1.7 (AMASS Solution Benchmarking). The tables Table 1, Table 2, Table 3, and Table 4 provide a rough estimation for the most relevant metric improvements (*'Target'* column), with the sole purpose of providing an indication of the expected progress regarding the estimated current state of practice (*'Current'* column).

For improvement assessment, we have specified qualitative indicators and associated quantitative values to them as follows: None – 0%; Very low – 10%; Low – 30%; Medium – 50%; High – 70%; Very High – 90%; Full – 100%. This way of characterising the current and target situations allows us to (1) estimate improvement quantitatively based on qualitative information, (2) mitigate uncertainties from only and directly using quantitative values (e.g., due to insufficient information about the current status), and (3) provide indicators that represent CPS overall assurance and certification in all the application domains addressed in AMASS.

Finally, the envisioned targets of the overall goals have been derived from the average improvement of their indicators.

Example: The target for G1 is 50%; $\frac{40+20+50+80+80+40+60+20+40}{9} = 47.7$ (approximately 50).

As indicated above, the metrics and their measurement will be refined during AMASS execution for each case study. For example, each metric could have a different weight for the assessment of the corresponding overall goal. This has not been taken into account yet because the weight is case study-, domain-, and even AMASS results-dependent, thus the necessary further analysis is not possible at the current project status. A more general approach has been used as a starting point and for providing an initial estimation of the improvements that AMASS can enable.

The Figure 3, Figure 4 and Figure 5 show at high-level how the four AMASS goals and their associated objectives are mapped to assessment questions. In the lowest level of the figure, the mapping from assessment questions to corresponding metrics is showed. The description of individual questions and metrics are in the following tables.

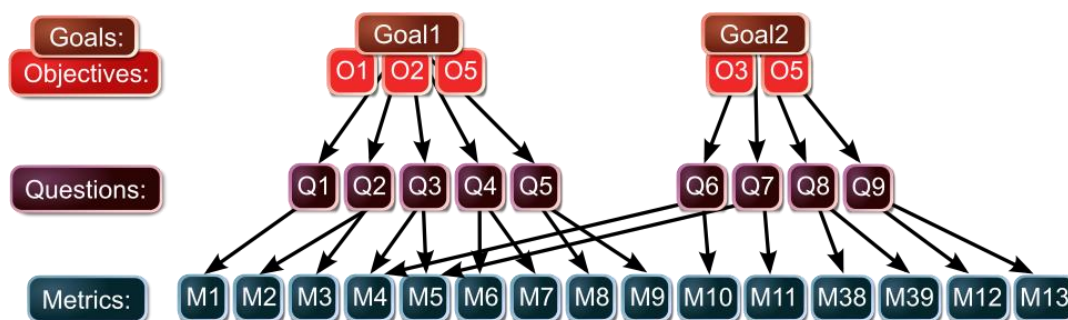


Figure 3. AMASS Goal 1 and Goal 2 mapped to Questions that are mapped to Metrics

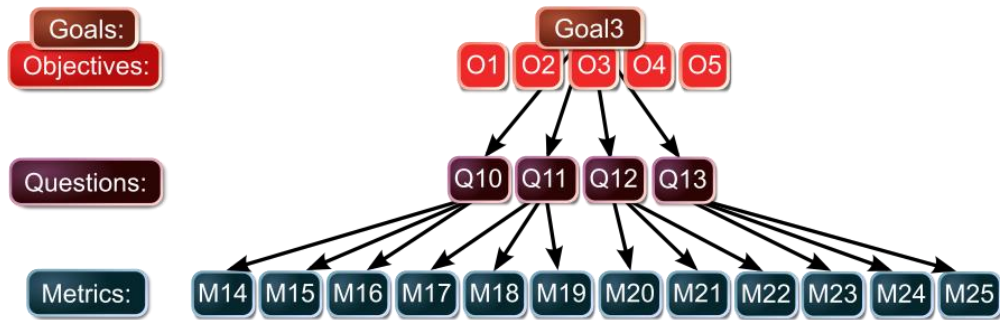


Figure 4. AMASS Goal 3 mapped to Questions that are mapped to Metrics

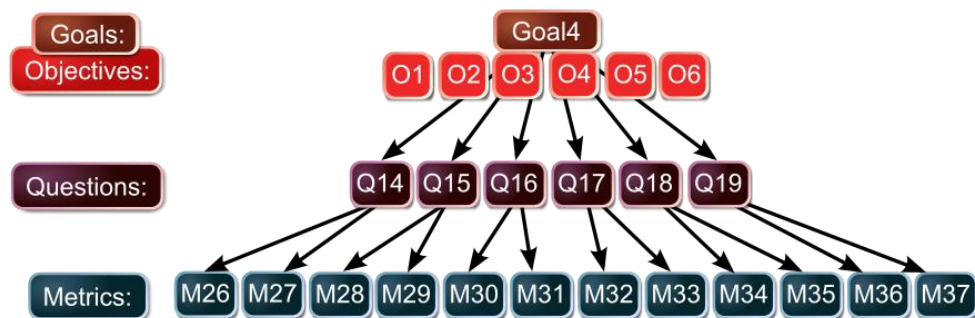


Figure 5. AMASS Goal 4 mapped to Questions that are mapped to Metrics

Table 1. Assessment of AMASS goal G1

G1: to demonstrate a potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%.				
AMASS objectives addressing this goal: O1, O2, O5		STOs addressing this goal: STO1 and STO2		
Overall rationale for the improvement: AMASS will reduce the execution time of assurance and certification/qualification activities by means of the AMASS Tool Platform, which will increase the level of automation of the activities and allow early detection of assurance issues. It must be noted that most assurance and certification/qualification activities cannot be fully automated. Human agents usually have to make some decisions (e.g., if a piece of safety evidence is adequate for a given assurance claim).				
Aspects addressed: Effort, assurance result reuse.				
Question	Basis	Metric	Current	Target
Q1: How can the effort for architecture-driven and multi-concern assurance be automated?	AMASS will reduce effort needed for architecture-driven and multi-concern assurance by automation of the activities. The prerequisite is having requirements, system architecture or system models formally specified to enable assurance automation.	M1: Automated architecture-driven and multi-concern assurance	Very low	Medium (40% gain)
Q2: How can the effort for determining the needs of architecture-driven and multi-concern assurance and certification/qualification be reduced?	AMASS will facilitate the determination of architecture-driven and multi-concern assurance needs and certification/qualification needs by providing means that enable a systematic analysis of such needs and provide (semi-)automated support. Similar means are scarce nowadays, especially for multi-concern assurance.	M2: Identification of consequences of CPS architecture on assurance and on certification/qualification	Medium	Low (20% decrease)
		M3: Identification of consequences of having to address several dependability aspects	High	Low (50% decrease)
Q3: How can the effort for documenting architecture-driven and multi-concern assurance, and for documenting architecture-driven and multi-concern certification/qualification be reduced?	Many architectural and multi-concern system aspects are very recent. Therefore, limited means exist nowadays for facilitating the reuse of their assurance results and their certification/qualification results (e.g., by a systematic characterization and structuring), and thus for exploiting the benefits in terms of effort reduction that the reuse can enable.	M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused	Very low	Very high (80% gain)
		M5: Multi-concern assurance results and multi-concern certification/qualification results reused	Very low	Very high (80% gain)
Q4: How can the effort for identifying issues in architecture-driven and multi-concern assurance	AMASS will greatly facilitate the identification of assurance risks thanks to the development of a systematic approach that explicitly supports the mechanisms necessary for such identification: management of links/traces between architecture and system	M6: Identification of architecture-based assurance risks	High	Low (40% decrease)
		M7: Identification of multi-concern-	Very high	Low

be reduced?	models, structured arguments, safety & security co-assurance, etc.	based assurance risks		(60% decrease)
Q5: What impact can the early identification of the above issues have in design efficiency?	As AMASS will allow practitioners to identify assurance risks early in the system design, and the sooner the risks are identified the lower the effort for addressing them. Project results can significantly contribute to effort reduction for addressing assurance risks.	M8: Addressing architecture-based assurance risks	High	Medium (20% decrease)
		M9: Addressing multi-concern-based assurance risks	Very high	Medium (40% decrease)

Table 2. Assessment of AMASS goal G2

G2: to demonstrate a potential reuse of assurance results (qualified or certified before), leading to 40% of cost reductions for component/product (re)certification/qualification activities.				
AMASS objectives addressing this goal: O3, O5		STOs addressing this goal: STO1, STO2, STO4		
Overall rationale for the improvement: AMASS will strongly focus on possible reuse of assurance results from different perspectives. These perspectives do not only relate to cross-domain assurance (STO4), but also deal with the reuse of information about architecture-driven and multi-concern assurance (e.g., via argumentation patterns). We analyse reuse based on assurance results (e.g., artefacts to create; aka assurance assets). This information can be turned into a cost analysis by specifying the cost associated to each assurance result.				
Aspects addressed: Assurance assets, argumentation elements, compliance justifications.				
Question	Basis	Metric	Current	Target
Q6: What is the impact of reusing architecture-driven assurance results?	AMASS will deal with the assurance of architectural aspects for which little support exists nowadays towards systematically analysing and documenting their assurance results. An adequate characterization of these results is a prerequisite for reuse approaches.	M4: Architecture-driven assurance results reused	Low	High (40% gain)
		M10: Assurance needs met after architecture-driven assurance reuse	Low	High (40% gain)
Q7: What is the impact of reusing multi-concern assurance results?	There is a growing interest in multi-concern assurance in both industry and academia, but the present means for addressing it are far from meeting many industrial needs (e.g., regarding systematic reuse). Co-assurance of safety and security concerns is of utmost relevance, and AMASS will pay great attention to it.	M5: Multi-concern assurance results reused	Very low	High (60% gain)
		M11: Assurance needs met after multi-concern assurance results reuse	Low	Medium (20% gain)
Q8: What is the impact of reusing certification/qualification results?	AMASS will deal with documented certification/qualification results. An adequate characterization of these results is a prerequisite for reuse approaches.	M38: Certification/qualification results reused	Low	High (40% gain)
		M39: Certification/qualification needs met after certification/qualification results reuse	Low	High (40% gain)
Q9: What is the impact of cross-domain reuse of assurance results?	Although some approaches have been developed recently for cross-domain assurance reuse (e.g., in OPENCOSS as well as in SafeCer), the current solutions must be extended and improved to better determine cross-domain reuse consequences and thus support a larger basis for reuse.	M12: Assurance results reused across domains	Very low	Medium (40% gain)
		M13: Assurance needs met after results reuse across domains	Very low	Medium (40% gain)

Table 3. Assessment of AMASS goal G3

G3: to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification risks of new safety/security-critical products.				
AMASS objectives addressing this goal: O1, O2, O3, O4, O5		STOs addressing this goal: STO1, STO2, STO3, STO4		
Overall rationale for the improvement: A very important aspect for AMASS is that assurance and certification efficiency should not jeopardize assurance information quality. Further, AMASS aims to increase the quality by providing a solution that allows practitioners to (1) detect new assurance and certification risks (those resulting from CPS facets, such as co-assurance of safety and security), (2) detect common risks more efficiently (e.g., the fact that information integrated from different tools does not meet standards’ requirements), and (3) avoid certain risks by following a more systematic and methodical approach for CPS assurance and certification. AMASS systematic supporting approach for CPS assurance and certification will mitigate the risks and might lead to discover potential new risks.				
Aspects addressed: Knowledge deficiencies, issues not identified, issues not analysed, issues not reported.				
Question	Basis	Metric	Current	Target
Q10: How can architecture-driven assurance contribute to the reduction of assurance and certification risk?	In the scope of architecture-driven assurance, the risks addressed in AMASS are related to aspects such as missing information of the correspondence between assurance and system models, gaps between components’ information and assurance needs, and inadequate architecture-based argumentation.	M14: Identified risks related to architecture-driven assurance	Medium	Very high (40% gain)
		M15: Mitigated risks related to architecture-driven assurance	Very low	Medium (40% gain)
		M16: Discovered unknown risks related to architecture-driven assurance	None	Low (30% gain)
Q11: How can multi-concern assurance contribute to the reduction of assurance and certification risk?	In the scope of multi-concern assurance in AMASS, risks can arise because dependencies and trade-offs among concerns are properly addressed and because of the lack of confidence in multi-concern argumentation.	M17: Identified risks related to multi-concern assurance	Very low	High (60% gain)
		M18: Mitigated risks related to multi-concern assurance	Very low	Medium (40% gain)
		M19: Discovered unknown risks related to multi-concern assurance	None	Low (30% gain)
Q12: How can seamless interoperability contribute to the reduction of assurance and certification risk?	For seamless interoperability, AMASS will mainly deal with risks related to those assurance needs that are not fulfilled or might not be fulfilled as a result of unsuitable information integration: between engineering tools, from stakeholders’ collaboration, about tools (qualified or not), etc.	M20: Identified risks related to seamless interoperability	Low	High (40% gain)
		M21: Mitigated risks related to seamless interoperability	Low	Medium (20% gain)
		M22: Discovered unknown risks related to seamless interoperability	None	Low (30% gain)
Q13: How can cross-domain assurance	AMASS will facilitate the identification of cross-domain assurance risks, which include insufficient awareness of the differences and	M23: Identified risks related to cross-domain assurance	Low	High (40% gain)

contribute to the reduction of assurance and certification risk?	commonalities between assurance practices in different domains, and of cross-domain development/reuse consequences.	M24: Mitigated risks related to cross-domain assurance	Very low	Medium (40% gain)
		M25: Discovered unknown risks related to cross-domain assurance	None	Low (30% gain)

Table 4. Assessment of AMASS goal G4

G4: to demonstrate a potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification technologies by 60%.				
AMASS objectives addressing this goal: O1, O2, O3, O4, O5, O6		STOs objectives addressing this goal: STO1, STO2, STO3, STO4		
Overall rationale for the improvement: AMASS explicitly deals with CPS assurance and certification for several application domains, which are some of the most relevant ones currently (e.g., there is a growing concern worldwide about automotive assurance due to recent safety and security incidents). AMASS also addresses cross-domain assurance and the assurance of different technologies (e.g., via architecture-driven and multi-concern assurance). Therefore, the provision of a common, harmonized solution for assurance based on different technologies and intended for different domains is inherent to AMASS. Seamless interoperability as well as the fostering of the AMASS community will also contribute to a sustainable impact.				
Aspects addressed: Assurance needs, assurance support means, stakeholders.				
Question	Basis	Metric	Current	Target
Q14: How can architecture-driven assurance contribute to sustainable impact?	The provision of common means for architecture-driven assurance to meet different needs will be shown in the case studies and will demonstrate that AMASS provides a harmonized, interoperable solution.	M26: Common means for architecture-driven assurance	Very low	Very high (80% gain)
		M27: Common architecture-driven assurance needs met	Low	High (40% gain)
Q15: How can multi-concern assurance contribute to sustainable impact?	Assurance of a diverse set of concerns will be supported in several case studies. In other words, AMASS results for multi-concern assurance will be able to address this area in a variety of scenarios for several application domains.	M28: Common means for multi-concern assurance	Very low	Very high (80% gain)
		M29: Common multi-concern assurance needs met	Very low	High (60% gain)
Q16: How can seamless interoperability contribute to sustainable impact?	Harmonization and interoperability are evidently inherent to seamless interoperability. AMASS will develop a solution that will facilitate the collaboration of tools and stakeholders in heterogeneous situations for which insufficient support is available today.	M30: Common means for seamless interoperability	Low	Very high (60% gain)
		M31: Assurance result types with seamless interoperability support	Low	Very high (60% gain)
Q17: How can cross-domain assurance contribute to sustainable impact?	The development and implementation of a cross-domain assurance solution implicitly requires that AMASS envisions a new framework and new tool support that harmonize assurance practices among domains and allow the domains to exchange assurance results.	M32: Common means for cross-domain assurance	Low	High (40% gain)
		M33: Common cross-domain assurance needs met	Low	Very high (60% gain)
Q18: How can AMASS eco-system contribute to sustainable impact?	The creation and later maintenance of the AMASS eco-system will ensure that AMASS results will be available as open source and will evolve in the future (probably within Polarsys and based on OPENCOS tooling).	M34: Open assurance and certification tool users	Very low	Medium (40% gain)
		M35: Open assurance and certification features	Low	High (40% gain)

Q19: How can AMASS community contribute to sustainable impact?	AMASS will build and foster its community, consisting of different supporting organizations and individuals (aka contributors). The existence of such a community guarantees a long lifecycle of AMASS results, beyond the project's duration.	M36: Open assurance and certification tool supporting organizations	Very low	High (60% gain)
		M37: Open assurance and certification tool contributors	Very low	High (60% gain)

2.3 Software Related Metrics

Since the AMASS Integration Framework is based on software tools, general software related metrics will be used to contribute to AMASS metrics measurement. E.g. the metric *MC10.5: Pages of documentation written per programmer month* can be used to provide information relevant to the question *Q3: How can the effort for documenting architecture-driven and multi-concern assurance, and for documenting architecture-driven and multi-concern certification/qualification be reduced?* that is in turn used to indicate whether the goal *G1: to demonstrate a potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%* was achieved.

The measurement of some metrics could be automated and the AMASS evaluation framework will recommend which metrics capture should be implemented in the AMASS Integration Framework.

Examples of common software metrics can be found in many books (e.g. [4], [5]):

1. Number of artefacts (requirements, components, objects)
2. Effort (time, cost, cost of poor quality)
3. Number of defects (introduced, detected, removed)
4. Changes (change requests, development cycles)
5. Product size, complexity
6. Process quality (efficiency, scalability, effectiveness, predictability)
7. Estimation accuracy
8. Productivity
9. Product quality
10. Customer satisfaction
11. Stability
12. Non-conformity

2.4 Common Evaluation Framework

This deliverable should provide a common evaluation framework for all case studies, which have very diverse goals. Therefore, the evaluation framework has to be very flexible.

For example, the case studies that do not aim to evaluate reusability will not measure any metric derived from AMASS goal G2 (to demonstrate a potential reuse of assurance results).

Based on the experience gained from prior projects (e.g. iFEST [11] and Crystal [12]), where we have observed that manually measured metrics have lower accuracy than automatically measured metrics due to human error, we recommend to automatically measure most of the metrics. For example, when measuring time spent on authoring requirements, the automatic measurement of time when a requirements document is opened or when a specific requirement is being edited or viewed is very imprecise. Yet it is usually more accurate than in the case when the system engineers are requested to fill how much time they spend on the tasks. Nevertheless, this might assume a certain behaviour of the engineers – not keep files open when doing some parallel task etc., and not working on local copies offline. Such constraints have to be carefully considered in the context of given circumstances. The best approach is to measure both automatically and manually and cross check both measurements.

3. AMASS Metrics

This section lists the metrics relevant to the evaluation of the AMASS results that will be measured in more than one case study. The purpose is to unify the terminology and its meaning. All the project partners that use a certain metric should have the same understanding of its measurement and e.g. its qualitative values, like e.g. the values “ASAP”, “Soon”, “Whenever” in the measurement of urgency. A similar goal of the unification of understanding and measurement was followed e.g. in [8], where the definitions of some software project-related metrics (effort, productivity, predictability, etc.) and examples of measured data can be found. The following four subsections define metrics that are derived from the four AMASS goals as described in the previous chapter.

The proposed metrics contain the comparison of two values, where one value is measured/estimated in the absence of the AMASS platform support, and the other value is measured when the AMASS platform is applied to the CPS development. In order to obtain well-defined comparisons, the same scope of assurance should be specified for both settings (with/without AMASS support). For example, the first iterations of the assurance could take into account only the requirements, architecture and design, and perform the assurance on an abstract level, maybe by considering a virtual system. As the system development advances, the scope of assurance is extended to the various levels of tests (unit, integration, system, acceptance) on the system implementation. The scope of the activities measured in the two compared values should always correspond.

The identifiers of individual metrics follow these rules:

M<number> is the form of the identifiers of common metrics (Chapter 3), e.g. M27.

MW<number1>.<number2> is the form of identifiers of WP-related metrics (Chapter 4), where *number1* is the number of the Work Package, e.g. MW3.6.

MC<number1>.<number2> is the form of identifiers of CS-related metrics (Chapter 5), where *number1* is the number of the Case Study, e.g. MC01.10.

3.1 Reducing Assurance and Certification/Qualification Effort

Demonstration of a potential reduction of assurance and certification/qualification effort by 50% requires significant increase of the level of automation of the activities as well as early defect detection.

The following subsections present the five questions formulated for goal G1 and the metrics that measure the corresponding reduction.

3.1.1 Automation of Architecture-Driven and Multi-Concern Assurance

Question 1: How can the effort for architecture-driven and multi-concern assurance be automated?

Metric 1: **Automated architecture-driven and multi-concern assurance**

$$\frac{\text{Automated assurance objectives}}{\text{Total assurance objectives}}$$

Description: The metric calculates the ratio of automated assurance effort (measured in person-time or cost) versus the total assurance effort (as if no automation was performed). It will measure actual assurance effort reduction by automating or semi-automating some part of the assurance process, rather than the ratio of automated assurance process items. By automating different assurance processes a different effort reduction is achieved. The range of this metric is [0 .. 1] or [0% .. 100%].

3.1.2 Identification of the Needs of Architecture-Driven and Multi-Concern Assurance

Question 2: How can the effort for determining the needs of architecture-driven and multi-concern assurance be reduced?

Metric 2: **Automated identification of consequences of CPS architecture on assurance**

$$\frac{\text{Automatically identified architecture consequences}}{\text{Total identified architecture consequences}}$$

Description: The metric calculates the ratio of automatically identified consequences of CPS architecture on assurance, versus the total identified consequences effort.

Examples of architectural features that can have consequences on assurance: Some kinds of architectural elements or patterns (e.g. fault tolerance, time/space partitioning, multicore technology, etc.) may be connected (e.g. via an appropriate standard) to the need for using a particular assurance pattern. The metric shows how significant is the amount of the auto-generated assurance cases compared to the amount of all (manually- and auto-generated) assurance cases.

Metric 3: **Automated identification of consequences of having to address several dependability aspects**

$$\frac{\text{Automatically identified dependability elements}}{\text{Total identified dependability elements}}$$

Description: The metric calculates the ratio of automatically identified consequences of having to address several dependability aspects, versus the total identified consequences. An overview of dependability is provided in [10].

The auto-generation of assurance cases should be parameterized by various types of requirements. If the focus is on the security related requirements, the corresponding assurance should be different from the assurance needs generated for the safety requirements. From a broader perspective, the generation does not need to be based only on the dependability requirements, but also on the mission requirements describing the function, behaviour, and performance, or the developmental requirements concerned with features like modifiability or extensibility.

3.1.3 Architecture-Driven and Multi-Concern Assurance Reuse

Question 3: How can the effort for documenting architecture-driven and multi-concern assurance be reduced?

Metric 4: **Architecture-driven assurance results reused**

$$\frac{\text{Reused assurance results}}{\text{Total assurance results}}$$

Description: The metric calculates the ratio of reused architecture-driven assurance results from different systems, to the total architecture-driven assurance results for the target system.

This measurement assumes that:

- There exist assurance results already, either in the form of complete instances of assurance cases for a library of components, or in the form of assurance case templates for some architectural patterns.
- Some of the corresponding components or architectural patterns can be reused in the new system.

The second assumption could be satisfied e.g. when a new system is an evolved version of an old assured system.

Metric 5: Multi-concern assurance results reused

$$\frac{\text{Reused assurance results}}{\text{Total assurance results}}$$

Description: The metric calculates ratio of reused multi-concern assurance results for two different systems, out of the total multi-concern assurance results for the same systems.

The reused assurance artefacts stem from the requirements that are related to the selected concern.

3.1.4 Architecture-Driven and Multi-Concern Assurance Risks

By assurance risk we understand the conditions that can make a product developer incapable of: (1) developing a system that complies with safety standards; (2) collecting and maintaining evidence to demonstrate safety, and; (3) making a third-party (e.g., a certification authority) gain confidence in the safe operation of a system, see [2].

Question 4: How can the effort for identifying issues in architecture-driven and multi-concern assurance be reduced?

Metric 6: Identification of architecture-based assurance risks

$$\frac{\text{Automatically identified assurance risks}}{\text{Total identified assurance risks}}$$

Description:

Automatically identified assurance risks = the number of assurance risks identified automatically on the base of the presence of architectural elements or architectural patterns.

Total identified assurance risks = the number of all assurance risks bound to architectural elements or architectural patterns.

Metric 7: Identification of multi-concern-based assurance risks

$$\frac{\text{Automatically identified assurance risks}}{\text{Total identified assurance risks}}$$

Description:

Automatically identified assurance risks = the number of assurance risks identified automatically upon processing the requirements related to a specific concern.

Total identified assurance risks = the number of all assurance risks identified manually and automatically by analysing the requirements related to a specific concern.

3.1.5 Early Identification Impact

Question 5: What impact can the early identification of the above issues have on design efficiency?

Metric 8: Addressing architecture-based assurance risks

$$\frac{\text{Cost of addressing assurance risks}}{\text{Cost of consequences of not addressing the risks}}$$

Description:

Cost of addressing assurance risks = Effort consumed at mitigation of identified architecture-related risks.

Cost of consequences of not addressing the risks = if the assurance risks related to a given architecture element/pattern were not addressed, would there be a rework needed later on? If so, estimate the cost of the rework.

Metric 9: Addressing multi-concern-based assurance risks

$$\frac{\text{Cost of addressing assurance risks}}{\text{Cost of consequences of not addressing the risks}}$$

Description:

Cost of addressing assurance risks = Effort consumed at mitigation of identified concern-related risks.

Cost of consequences of not addressing the risks = if the assurance risks related to a given concern were not addressed, would there be a rework needed later on? If so, estimate the cost of the rework.

3.2 Reusing Assurance Results

Demonstration of a potential reuse of assurance results that were certified or qualified before to reach 40% cost reduction requires multiple perspectives of reuse to be deployed; e.g. cross-domain assurance reuse and reuse of information about architecture-driven and multi-concern assurance (e.g., via argumentation patterns), among others.

The aspects of being architecture-driven, being multi-concern, and being cross-domain are independent, “orthogonal”. This means that an assurance result might be certified or qualified independently because of any of these three aspects. Therefore, it is possible that the same assurance result could be counted as *Reused assurance result* independently in any of the following subsections. The number *Total assurance results* should be the same for all three – the architecture-driven, multi-concern, and cross-domain related metrics.

The following subsections presents 3 questions formulated for the goal G2 and metrics that measure the answered reduction.

3.2.1 Reusing Architecture-Driven Assurance Results

Question 6: What is the impact of reusing architecture-driven assurance results?

Metric 4: **Architecture-driven assurance results reused**

$$\frac{\text{Reused assurance results}}{\text{Total assurance results}}$$

Description: The metric calculates the ratio of reused architecture-driven assurance results from different systems, to the total architecture-driven assurance results of the target system.

For a given project, there is a set of architecture-driven assurance results (i.e. results of assurance cases that were auto-generated from architectural elements or patterns). Some of these assurance results were certified/qualified before. The *Reused assurance results* metric provides the number of those results, whose certification or qualification could be reused without significant additional effort. The *Total assurance results* indicates the number of all assurance results of the project.

Metric 10: **Assurance needs met after architecture-driven assurance reuse**

$$\frac{\text{Assurance needs met}}{\text{Total assurance needs}}$$

Description: The *Total assurance needs* is the number of all the requirements that have to be covered by an assurance result, e.g. the number of safety requirements. The *Assurance needs met* is the number of those assurance related requirements whose assurance was found outside of the given project on the base of the architecture-driven reuse.

3.2.2 Reusing Multi-Concern Assurance Results

Question 7: What is the impact of reusing multi-concern assurance results?

Metric 5: Multi-concern assurance results reused

$$\frac{\text{Reused assurance results}}{\text{Total assurance results}}$$

Description: The metric calculates the ratio of reused multi-concern assurance results for two different systems, to the total multi-concern assurance results for the same system.

For a given project, there is a set of multi-concern assurance results (i.e. results of assurance cases, that were copied from the results obtained for assessment of a different concern). Some of these assurance results were certified/qualified before. The *Reused assurance results* metric provides the number of those results, whose certification or qualification could be reused without significant additional effort. The *Total assurance results* indicates the number of all assurance results of the project.

Metric 11: Assurance needs met after multi-concern assurance results reuse

$$\frac{\text{Assurance needs met}}{\text{Total assurance needs}}$$

Description: The *Total assurance needs* is the number of all the requirements that have to be covered by an assurance result, e.g. the number of safety requirements. The *Assurance needs met* is the number of those assurance related requirements whose assurance was found outside of the given project on the base of the multi-concern reuse.

3.2.3 Reuse of Certification and Qualification Results

Question 8: What is the impact of reusing certification/qualification results?

Metric 38: Certification and qualification results reused

$$\frac{\text{Reused certification and qualification results}}{\text{Total certification and qualification results}}$$

Description: The metric calculates the ratio of reused certification and qualification results for two different systems, to the total certification and qualification results for the same systems.

For a given project, there is a set of certification and qualification results present in some other domain. Some of these certification and qualification results were obtained before in the other domain. The *Reused certification and qualification results* metric provides the number of those results, whose certification or qualification could be reused without significant additional effort. The *Total certification and qualification results* indicates the number of all certification and qualification results of the project.

Metric 39: Certification and qualification needs met after results reuse

$$\frac{\text{Certification and qualification needs met}}{\text{Total certification and qualification needs}}$$

Description: The *Total certification and qualification needs* is the number of all the requirements that have to be covered by an certification and qualification result, e.g. the number of safety requirements. The *Certification and qualification needs met* is the number of those certification and qualification related requirements whose assurance was found outside of the given project on the base cross-domain reuse.

3.2.4 Cross-Domain Reuse of Assurance Results

Question 9: What is the impact of cross-domain reuse of assurance results?

Metric 12: Assurance results reused across domains

$$\frac{\text{Reused assurance results}}{\text{Total assurance results}}$$

Description: The metric calculates the ratio of reused multi-concern assurance results for two different systems, to the total multi-concern assurance results for the same systems.

For a given project, there is a set of assurance results present in some other domain. Some of these assurance results were certified/qualified before in the other domain. The *Reused assurance results* metric provides the number of those results, whose certification or qualification could be reused without significant additional effort. The *Total assurance results* indicates the number of all assurance results of the project.

Metric 13: Assurance needs met after results reuse across domains

$$\frac{\text{Assurance needs met}}{\text{Total assurance needs}}$$

Description: The *Total assurance needs* is the number of all the requirements that have to be covered by an assurance result, e.g. the number of safety requirements. The *Assurance needs met* is the number of those assurance related requirements whose assurance was found outside of the given project on the base cross-domain reuse.

3.3 Reducing Assurance and Certification and Qualification Risks

By assurance, certification and qualification risk we understand the conditions that can make a product developer incapable of: (1) developing a system that complies with safety standards; (2) collecting and maintaining evidence to demonstrate safety, and; (3) making a third-party (e.g., a certification authority) gain confidence in the safe operation of a system, see [2].

Demonstration of a potential 35% reduction of assurance and certification/qualification risks of new safety-critical products requires innovated technology that preserves assurance information quality.

The following subsections present 4 questions formulated for goal G3 and metrics that measure the required reduction.

If the probability and severity of individual risks is available, the following metrics dealing with risks can take into the account these attributes and focus on just a selected set of risks with selected values of these attributes.

The following subsections use the terms identified risks, discovered unknown risks, and total risks that are explained here:

Identified risks are risks that were found automatically by the AMASS platform. These identified risks can be divided into two disjoint groups: the risks that were/could be found also without using the AMASS platform, and the (automatically) *discovered unknown risks*. The quantity *total risks* means all the risks together, those identified automatically (which include the discovered unknown risks) and those found by other means.

3.3.1 Reducing Risks by Architecture-Driven Assurance

Question 10: How can architecture-driven assurance contribute to the reduction of assurance and certification risk?

Metric 14: Identified risks related to architecture-driven assurance

$$\frac{\text{Correctly identified risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks automatically identified based on architecture-driven assurance, out of all the risks.

Metric 15: Mitigated risks related to architecture-driven assurance

$$\frac{\text{Mitigated risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks that were mitigated due to the architecture-driven assurance out of all the risks that would need to be addressed when the architecture-driven assurance was not performed. Some risks might be e.g. mitigated by changing the Development Assurance Level of the given aircraft function by means of a change in the architecture.

Metric 16: **Discovered unknown risks related to architecture-driven assurance**

$$\frac{\text{Newly discovered risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks newly discovered based on architecture-driven assurance out of all the risks.

3.3.2 Reducing Risks by Multi-Concern Assurance

Question 11: How can multi-concern assurance contribute to the reduction of assurance and certification risk?

Metric 17: **Identified risks related to multi-concern assurance**

$$\frac{\text{Correctly identified risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks identified based on multi-concern assurance out of all the risks.

Metric 18: **Mitigated risks related to multi-concern assurance**

$$\frac{\text{Mitigated risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of risks mitigated based on multi-concern assurance out of all the risks.

Metric 19: **Discovered unknown risks related to multi-concern assurance**

$$\frac{\text{Newly discovered risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of risks newly discovered based on multi-concern assurance out of all the risks.

3.3.3 Reducing Risks by Seamless Interoperability

Interoperability in our context is the possibility of the communication between the tools and other components of the given development (and assurance) environment. One tool can invoke functionalities of another tool. Data produced and exported by one tool are understood by the tool that consumes the data. The term *seamless* points to the absence of the loss of information as it flows among the tools, and to the perception of the cooperating tools by their users. From a broader perspective the interoperability pertains not only to the tools, but also to the development and assurance, certification and qualification processes. In other words, the employed technologies can cooperate as if they were tightly connected via a common language.

Question 12: How can seamless interoperability contribute to the reduction of assurance and certification risk?

Metric 20: **Identified risks related to seamless interoperability**

$$\frac{\text{Correctly identified risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks identified based on seamless interoperability that actually occur versus all the risks.

Metric 21: **Mitigated risks related to seamless interoperability**

$$\frac{\text{Mitigated risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of risks mitigated based on seamless interoperability out of all the risks.

Metric 22: **Discovered unknown risks related to seamless interoperability**

$$\frac{\text{Newly discovered risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of risks newly discovered based on seamless interoperability versus all the risks.

3.3.4 Reducing Risks by Cross-Domain Assurance

Question 13: How can cross-domain assurance contribute to the reduction of assurance and certification risk?

Metric 23: **Identified risks related to cross-domain assurance**

$$\frac{\text{Correctly identified risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks identified based on cross-domain assurance out of all the risks.

Metric 24: **Mitigated risks related to cross-domain assurance**

$$\frac{\text{Mitigated risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks mitigated based on cross-domain assurance out of all the risks.

Metric 25: **Discovered unknown risks related to cross-domain assurance**

$$\frac{\text{Newly discovered risks}}{\text{Total risks}}$$

Description: The metric is calculated as the ratio of the risks newly discovered based on cross-domain assurance out of all the risks.

3.4 Sustainable Impact by Harmonization and Interoperability

Demonstration of a potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification tool technologies by 60% requires the presence of a common metamodel and well-defined interdependencies of the assurance and certification-related artefacts. It also requires that the communication paths between the tools are established, which enable appropriate sharing and “composition” of the data. 60% harmonization means that such portion of the assurance and certification/qualification technologies (tools and procedures observed at an appropriate level of granularity) can be mapped one-to-one between domains. The harmonized part of the technologies

can be perceived as a universally applicable backbone of the assurance and certification/qualification activities.

The following subsections presents 6 questions formulated for goal G4 and metrics that measure the answered reduction. By the term *common means* we understand the generally applicable tools and procedures (identified at an appropriate level of granularity) that contribute to or support the given process/activity.

3.4.1 Sustainable impact by architecture-driven assurance

Question 14: How can architecture-driven assurance contribute to sustainable impact?

Metric 26: **Common means for architecture-driven assurance**

$$\frac{\text{Effort when common means are used}}{\text{Effort when common means are not used}}$$

Description: This metric expresses the reduction of the effort in performing the architecture-driven assurance when the AMASS technology is used as compared to the analogous process performed without the support of the AMASS technology. The value of 0 would correspond to the fully automated assurance, the value of 1 would correspond to no improvement, and a value greater than 1 would mean deterioration of the process.

Metric 27: **Common architecture-driven assurance needs met**

$$\text{Number of common needs met}$$

Description: This metric provides the number of requirements that are architecture-driven and that are common to CPS systems with the same architectural elements or patterns. The effort taken by the assurance of such requirements can be significantly lower because the similar or identical common assurance result is available already. The AMASS platform is expected to contribute to both: the identification of such needs and the assurance that these needs are met. Therefore, the number of satisfied assurance needs is an appropriate indicator of the impact of the technology.

3.4.2 Sustainable impact by multi-concern assurance

Question 15: How can multi-concern assurance contribute to sustainable impact?

Metric 28: **Common means for multi-concern assurance**

$$\frac{\text{Effort when common means are used}}{\text{Effort when common means are not used}}$$

Description: This metric expresses the reduction of the effort in performing the multi-concern assurance when the AMASS technology is used as compared to the analogous process performed without the support of the AMASS technology. The value of 0 would correspond to the fully shared assurance, and the value of 1 would correspond to no improvement, and a value greater than 1 would mean deterioration of the process.

Metric 29: **Common multi-concern assurance needs met**

$$\text{Number of common needs met}$$

Description: This metric provides the number of requirements that are common to multiple concerns. The effort taken by the assurance of such requirements can be significantly lower because the common assurance result is available once it is completed in the context of any of the related concerns. The AMASS platform is expected to contribute to both: the identification of such needs and the assurance that these needs are met. Therefore, the number of satisfied assurance needs is an appropriate indicator of the impact of the technology.

3.4.3 Sustainable impact by seamless interoperability

Question 16: How can seamless interoperability contribute to sustainable impact?

Metric 30: **Common means for seamless interoperability**

$$\frac{\text{Effort when common means are used}}{\text{Effort when common means are not used}}$$

Description: This metric expresses the reduction of the effort in performing the assurance when the AMASS technology with harmonized/interoperable tools is used as compared to the analogous process performed without the support of the AMASS technology. The value of 0 would correspond to fully shared assurance, and the value of 1 would correspond to no improvement, and a value greater than 1 would mean deterioration of the process.

Metric 31: **Assurance result types with seamless interoperability support**

$$\text{Number of assurance result types}$$

Description: This metric provides the number of assurance result types for which there is a seamless interoperability support. Various types of artefacts can be considered as evidence of assurance cases. Each such a type of artefact can be generated by zero, one, or more tools within the AMASS platform. "Result type with seamless interoperability support" means that there is a continuous path in the workflow diagram from the assurance needs to the assurance results. The impact of sharing a supported assurance result type is that the functionalities of the interoperating tools provide a broader foundation for processing and examining the assurance related aspects of the system, than the functionalities provided by just a single separated tool. Therefore, the number of such shared types is relevant to the impact of the technology.

3.4.4 Sustainable impact by cross-domain assurance

Question 17: How can cross-domain assurance contribute to sustainable impact?

Metric 32: **Common means for cross-domain assurance**

$$\frac{\text{Effort when common means are used}}{\text{Effort when common means are not used}}$$

Description: This metric expresses the reduction of the effort in performing the cross-domain assurance when the AMASS technology is used as compared to the analogous process performed without the support of the AMASS technology. The value of 0 would correspond to fully shared assurance, and the value of 1 would correspond to no improvement, and a value greater than 1 would mean deterioration of the process.

Metric 33: **Common cross-domain assurance needs met**

$$\text{Number of common needs met}$$

Description: This metric provides the number of requirements that are similar or identical in more than one domain. The effort taken by the assurance of such requirements can be significantly lower because the common assurance result can be shared across multiple domains. The AMASS platform is expected to contribute to both: the identification of such needs and the assurance that these needs are met. Therefore, the number of satisfied assurance needs is an appropriate indicator of the impact of the technology.

3.4.5 Sustainable impact by AMASS eco-system

Question 18: How can AMASS eco-system contribute to sustainable impact?

Metric 34: Open assurance and certification tool users

Number of tool users

Description: This metric expresses the number of individuals who perform assurance/certification related activities and while performing them they use one or more tools of the AMASS platform. For each such metric, it must be clarified which categories of users will be counted and what is the expected accuracy. These categories may differ at various stages of the AMASS project.

Metric 35: Open assurance and certification features

Number of features

Description: This metric provides the number of assurance- and certification-related features offered by the tools in the AMASS platform. Examples of such features are: inclusion of GSN diagram to the assurance case, (partial) generation of assurance case from system component specification, reporting of assurance case status, etc. The features can be categorised according to their importance (e.g. “success-critical”, “nice to have”). Enumerating the metric just for certain importance levels would yield more specific information on the possibilities of applications.

3.4.6 Sustainable impact by AMASS community

Question 19: How can AMASS community contribute to sustainable impact?

Metric 36: Open assurance and certification tool supporting organizations

Number of supporting organizations

Description: This metric will be measured by counting the number of different organizations listed on specified sources of information (e.g. some web-page similar to <http://www.amass-ecsel.eu/partners>). For each such metric it must be clarified which categories of organizations will be counted and what is the expected accuracy. These categories may differ at various stages of the AMASS project.

Metric 37: Open assurance and certification tool contributors

Number of contributors

Description: This metric provides the number of individual contributors who support the development or application of the AMASS platform.

3.5 Common Evaluation Procedures

The evaluation process is described schematically in Appendix A Evaluation – EPF Process Description. The results of the first two phases of Planning and Definitions are to a large extent presented in the current document. The outcomes of the consequent phases of Data Collection and of Interpretation shall constitute the content of other deliverables.

Common evaluation procedures are supported by the common evaluation framework described in the section 3.6 Common Evaluation Framework.

3.6 Common Evaluation Framework

For the sake of precision, objectivity and quick acquisition it is desirable to measure the values automatically.

The measured objects (products, processes, resources) are sometimes represented in the form of data managed by a tool. The aim is to use the tool’s capabilities to generate the measured values. If possible, it might be beneficial to add an extension to the tool that will perform the needed measurement

automatically. An example might be that Papyrus counts the number of components that do not have a related contract.

From the existing tools that support measurement and/or interpretation of data the following could be mentioned: JIRA, GitHub, Excel.

4. Technical Solution Metrics, Processes, and Tools

This chapter contains technical solutions' specific metrics that come from technical work packages: WP3, WP4, WP5, and WP6.

4.1 Metrics from WP3 – Architecture Driven Assurance

The initial metrics for the AMASS evaluation framework that could be measured from WP3 results:

G1: A potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%

- {Q1} MW3.1 percentage of (safety and security) requirements formalized (as contracts)
- {Q1} MW3.2 number of pieces of evidence and claims automatically generated (from contracts based design)
- {Q4} MW3.3 effort reduction for identifying impact on assurance case of different architectural design choices for implementation or changes
- {Q1} MW3.4 number of V&V activities automatically supported
- {Q3} MW3.5 number of applied architectural patterns

G2: to demonstrate a potential reuse of assurance results (qualified or certified before), leading to 40% of cost reductions for component/product (re)certification/qualification activities

- {Q3} MW3.6 percentage of pieces of evidence and claims from reused certified/qualified components
- {Q3} MW3.7 number of pieces of evidence and claims associated to applied architectural patterns

G3: to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification risks of new safety/security-critical products

- {Q1} MW3.8 percentage of requirements verified by V&V analysis (by using contracts based design approach)
- {Q5} MW3.9 percentage of reduction of system design errors (automatically discovered by using contract-based design approach)
- {Q5} MW3.10 percentage of reduction of components integration errors (automatically discovered by using contract-based design approach)

G4: to demonstrate a potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification technologies by 60%

- {Q18} MW3.11 number of languages and notations with which the AMASS system component/specification metamodel shares concepts
- {Q19} MW3.12 standardization level of languages and notations covered by AMASS

4.2 Metrics from WP4 – Multi-Concern Assurance

G1: A potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%

- {Q1} MW4.1 number of design iterations required when applying combined multi-concern engineering methods in relation to those needed with traditional separate treatment of concerns

- {Q1} MW4.2 reduction of effort for the re-generation of pieces of evidence after changing functional/non-functional requirements to the system by using a multi-concern-compliant workflow tool
- {Q1} MW4.3 number or share of automatically generated pieces of evidence (solutions) for multi-concern arguments
- {Q2} MW4.4 an estimation of time needed for separate safety and security engineering process and the co-engineering process

G2: to demonstrate a potential reuse of assurance results (qualified or certified before), leading to 40% of cost reductions for component/product (re)certification/qualification activities

- {Q3} MW4.5 number or share of multi-concern-related re-certification steps saved in the automated workflow by re-using certified components
- {Q7} MW4.6 reduction of effort for adapting the assurance case argumentation after changing functional/non-functional requirements to the system by using multi-concern argumentation
- {Q1} MW4.7 the percentage of [number of assurance case argumentation pattern] / [number of individual assurance case argumentation]

G3: to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification risks of new safety/security-critical products

- number or share of architectural/design modifications saved by
 - {Q2} MW4.8 combined safety/security co-engineering
 - {Q2} MW4.9 combined safety/performance co-engineering
 - {Q2} MW4.10 combined security/performance co-engineering
 - {Q2} MW4.11 combined safety/security/performance co-engineering
- number or share of requirements with relationships between different concerns
 - {Q4} MW4.12 requirements with “conflicting-impact” relationship
 - {Q3} MW4.13 requirements with “supporting-impact” relationship
 - {Q4} MW4.14 requirements with “dependency-impact” relationship
- {Q5} MW4.15 number or share of “conflicting-impact” relationships identified already in the analysis of the first system concept (and consequently avoiding iterations for later improvement of architecture/design and multi-concern-argumentation)

G4: to demonstrate a potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification technologies by 60%

- n/a

4.3 Metrics from WP5 – Seamless Interoperability

The initial metrics for the AMASS evaluation framework that could be measured from WP5 results:

G1: A potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%

- {Q16} MW5.1 Assurance information collection effort: actions to collect assurance information from external tools
- {Q16} MW5.2 Assurance information exchange effort: actions to exchange assurance information between different stakeholders

G4: A potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification tool technologies by 60%

- {Q18} MW5.3 Common collaboration means: number of technologies that can be applied to several collaboration scenarios
- {Q16} MW5.4 Tool interoperability domains: number of artefact types for which some tool interoperability means exist
- {Q18} MW5.5 Tool connectors: number of available tool connectors
- {Q16} MW5.6 Inter-connected tools: number of inter-connected tools
- {Q18} MW5.7 Standardised tool interoperability means: number of standardised or standard-based tool interoperability means
- {Q16} MW5.8 Interoperability level: level reached from using AMASS results; possible reference model candidates:
 - LISI Model: Levels Of Information Systems Interoperability (LISI) Reference Model [13], ISA² - Interoperability solutions for public administrations, businesses and citizens [14], Interoperability Maturity Mode [15], Interoperability requirements, TOGAF 9.1 [16], Interoperability Levels for Dublin Core Metadata [17], HIMSS interoperability standards [18], Understand the Three Levels of Interoperability [19], and Conceptual interoperability [20].

4.4 Metrics from WP6 – Cross/Intra Domain Reuse

Metrics that enable quantitative evaluation of the reuse are an extension of the metrics used in Product Line Engineering. The ‘components’ in a product line are replaced by ‘process elements’ in the context of process lines and by ‘assurance elements’ in the context of assurance cases. The metrics *Size of Commonality*, *Product-related Reusability* and *Relationship Ratio* apply to the goals G1 and G2. The metrics are concerned with cross/intra domain reuse measurement and are especially useful for objectively establishing a case for the setup of product/process/assurance lines. A comparison of the measurements of several product/process/assurance lines provides a means for prioritization of effort and allocation of resources based on their relative values. For more details, please see the deliverable D6.2 [21].

G1: A potential gain for design efficiency of complex CPS by reducing their assurance and certification/qualification effort by 50%

- {Q9} MW6.1 Size of Commonality (SoC_{SF}) – the number of reusable components in a product line determined by comparing the component signatures while factoring the impact of the product line input costs
- {Q9} MW6.2 Product-related Reusability (PrR_{SF}) – the extent of reusability of the common components for a specific product while factoring the impact of the product line input costs
- {Q9} MW6.3 Relationship Ratio (RR_{SF}) – the relationship between any two products in a product line while factoring the impact of the product line input costs

G2: to demonstrate a potential reuse of assurance results (qualified or certified before), leading to 40% of cost reductions for component/product (re)certification/qualification activities

- {Q9} MW6.1 Size of Commonality (SoC_{SF}) – the number of reusable components in a product line determined by comparing the component signatures while factoring the impact of the product line input costs
- {Q9} MW6.2 Product-related Reusability (PrR_{SF}) – the extent of reusability of the common components for a specific product while factoring the impact of the product line input costs
- {Q9} MW6.3 Relationship Ratio (RR_{SF}) – the relationship between any two products in a product line while factoring the impact of the product line input costs

G3: to demonstrate a potential raise of technology innovation led by 35% reduction of assurance and certification/qualification risks of new safety/security-critical products

- n/a

G4: to demonstrate a potential sustainable impact in CPS industry by increasing the harmonization and interoperability of assurance and certification/qualification technologies by 60%

- n/a

5. Case Study-Specific Metrics, Processes, and Tools

This chapter shows which common metrics derived from AMASS goals, technical solution metrics and specific case study metrics will be measured by individual case studies. The current vision will be reviewed in the future and thus might be modified.

This should be compliant with the AMASS goals that each case study will contribute to, as described in D1.1 [24]. From each AMASS goal that a given case study contributes to, the case study has to have at least one metric.

The following table provides an overall idea of used common and technical solution metrics of each case study:

Table 5. Metrics and their usage in the Case Studies

Metric	CS 1	CS 2	CS 3	CS 4	CS 5	CS 6	CS 7	CS 8	CS 9	CS 10	CS 11
Common metrics											
M1: Automated architecture-driven and multi-concern assurance	X			X	X		X	X		X	
M2: Identification of consequences of CPS architecture on assurance and on certification/qualification		X	X			X			X		
M3: Identification of consequences of having to address several dependability aspects								X			
M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused	X			X					X	X	
M5: Multi-concern assurance results and multi-concern certification/qualification results reused	X			X				X		X	
M6: Identification of architecture-based assurance risks			X	X		X	X		X	X	
M7: Identification of multi-concern-based assurance risks								X		X	
M8: Addressing architecture-based assurance risks			X			X	X		X	X	
M9: Addressing multi-concern-based assurance risks								X			
M10: Assurance needs met after architecture-driven assurance reuse		X	X	X		X				X	
M11: Assurance needs met after multi-concern assurance results reuse								X		X	
M12: Assurance results reused across domains		X									
M13: Assurance needs met after results reuse across domains											
M14: Identified risks related to architecture-driven assurance	X	X	X			X	X		X		
M15: Mitigated risks related to architecture-driven assurance			X			X	X		X		
M16: Discovered unknown risks related to architecture-driven assurance		X	X			X	X		X		
M17: Identified risks related to multi-concern assurance											
M18: Mitigated risks related to multi-concern assurance											
M19: Discovered unknown risks related to multi-concern assurance											
M20: Identified risks related to seamless interoperability			X				X				
M21: Mitigated risks related to seamless interoperability			X				X				

Metric	CS 1	CS 2	CS 3	CS 4	CS 5	CS 6	CS 7	CS 8	CS 9	CS 10	CS 11
M22: Discovered unknown risks related to seamless interoperability			X				X				
M23: Identified risks related to cross-domain assurance											
M24: Mitigated risks related to cross-domain assurance											
M25: Discovered unknown risks related to cross-domain assurance											
M26: Common means for architecture-driven assurance		X	X	X		X	X			X	
M27: Common architecture-driven assurance needs met											
M28: Common means for multi-concern assurance				X							
M29: Common multi-concern assurance needs met										X	
M30: Common means for seamless interoperability		X	X				X				X
M31: Assurance result types with seamless interoperability support	X	X	X				X				X
M32: Common means for cross-domain assurance											
M33: Common cross-domain assurance needs met		X									
M34: Open assurance and certification tool users											
M35: Open assurance and certification features											
M36: Open assurance and certification tool supporting organizations											
M37: Open assurance and certification tool contributors											
WP3 metrics											
percentage of (safety and security) requirements formalized (as contracts)		X	X		X	X	X			X	
number of pieces of evidence and claims automatically generated (from contracts based design)							X		X		
effort reduction for identifying impact on assurance case of different architectural design choices for implementation or changes								X			
number of V&V activities automatically supported	X	X	X		X	X	X		X	X	
number of applied architectural patterns							X				
percentage of pieces of evidence and claims from reused certified/qualified components											
number of pieces of evidence and claims associated to applied architectural patterns							X				
Percentage of requirements verified by V&V analysis (by using contracts based design approach).		X	X		X	X	X		X		
Percentage of reduction of system design errors (automatically discovered by using contracts based design approach).		X	X			X	X		X		
Percentage of reduction of components integration errors (automatically discovered by using contracts based design approach).											
number of languages and notations with which the AMASS system component/specification metamodel shares concepts											
standardization level of languages and notations covered by AMASS											
WP4 metrics											

Metric	CS 1	CS 2	CS 3	CS 4	CS 5	CS 6	CS 7	CS 8	CS 9	CS 10	CS 11
number of design iterations							X				
reduction of effort for the re-generation										X	
number or share of automatically generated pieces of evidence	X									X	
time needed for separate safety and security engineering process and the co-engineering process	X	X			X						
number or share of multi-concern-related re-certification steps										X	
reduction of effort for adapting the assurance case argumentation											
assurance case argumentation patterns to all individual assurance case argumentation											
architectural/design modifications saved by combined safety/security co-engineering	X										
architectural/design modifications saved by combined safety/performance co-engineering											
architectural/design modifications saved by combined security/performance co-engineering											
architectural/design modifications saved by combined safety/security/performance co-engineering											
requirements with relationships between different concerns with "conflicting-impact" relationship		X									
requirements with relationships between different concerns with "supporting-impact" relationship											
requirements with relationships between different concerns with "dependency-impact" relationship											
"conflicting-impact" relationships identified already in the analysis of the first system concept											
WP5 metrics											
Assurance information collection effort: actions to collect assurance information from external tools	X	X	X								X
Assurance information exchange effort: actions to exchange assurance information between different stakeholders		X	X								X
Common collaboration means: number of technologies that can be applied to several collaboration scenarios											
Tool interoperability domains: number of artefact types for which some tool interoperability means exists	X	X	X				X				
Tool connectors: number of available tool connectors			X				X				
Inter-connected tools: number of inter-connected tools			X				X				X
Standardised tool interoperability means: number of standardised or standard-based tool interoperability means		X	X				X				
Interoperability level: level reached from using AMASS results	X	X	X								X
WP6 metrics											
Size of Commonality											
Product-related Reusability	X										

Metric	CS 1	CS 2	CS 3	CS 4	CS 5	CS 6	CS 7	CS 8	CS 9	CS 10	CS 11
Relationship Ratio											
CS1 metrics											
Automation of architecture-driven safety and security assurance process for the RTU	X										
RTU compliance management effort	X										
Effort for determining the level of compliance of the RTU respect to the standards selected	X										
Effort for running safety/security analysis of the RTU	X										
Reuse of security and safety assurance results for other RTU platforms	X										
Security and safety assurance reuse in RTU upgrade	X										
Reusing architecture-driven assurance results for RTUs	X										
Reduce the effort for identifying safety and security assurance risks for RTUs	X										
Reduce the effort for identifying architecture-based assurance risks for RTUs	X										
Reduce compliance management risks and automated documentation	X										
Reducing risks by seamless interoperability of tools related to RTU development	X										
Interoperability with other tools related to RTU development.	X										
CS2 metrics											
Effort spent on assurance activities		X	X				X				X
Rate of detected and solved issues during test phases		X									X
Manual work leading to poor quality		X	X								X
CS3 metrics											
Effort spent on assurance activities			X				X				X
Rate of detected and solved issues during test phases											X
Reuse of contract based assurance											X
Manual work leading to poor quality			X								X
CS4 metrics											
M-CS4-1: Number of issues discovered during design phases				X						X	
M-CS4-2: Ration of Number of RAMS issues that differs after a system specification change				X						X	
CS5 metrics											
Effort spent on assurance activities					X		X				
Security defaults detected					X						
CS6 metrics											
Cost of formal proof versus functional tests						X					
Early detection of safety issues						X					
Assurance raise thanks to use of the approach						X					
Reducing qualification effort						X					
Automation of architecture driven assurance						X	X				
CS7 metrics											

Metric	CS 1	CS 2	CS 3	CS 4	CS 5	CS 6	CS 7	CS 8	CS 9	CS 10	CS 11
Effort Spent on Development Process							X				X
Cost of Poor Quality of Development Process							X				
Defect Introduced by Development Process							X				
Defect Detected by Development Process							X				
Defect Removed by Development Process							X				
CS8 metrics											
<i>There are no further metrics specific only to this case study</i>											
CS9 metrics											
Effort spent on assurance activities									X		
Number of functional issues discovered during design phases									X		
Number of safety issues discovered during design phases									X		
CS10 metrics											
Estimate number of bugs in the code from static analysis and from dynamic execution of the code.										X	
Estimate the number of future failures.										X	
Lines of source code written per programmer month.										X	
Object instructions produced per programmer month.										X	
Pages of documentation written per programmer month.										X	
Test cases written and executed per programmer month.										X	
Number of parameters										X	
Number of modules.							X			X	
Number of modules called (estimating complexity of maintenance).										X	
Data Bindings: Triplet (p, x, q), where p and q are modules and X is variable within scope of both p and q .										X	
Used data binding										X	
Actual data binding										X	
Cohesion metric										X	
Techniques for software cost estimation										X	
cost estimate										X	
CS11 metrics											
Effort spent on assurance activities							X				X
Rate of detected and solved issues during test phases											X
Reuse of contract based assurance											X
Manual work leading to poor quality											X

5.1 Case Study 1

Case Study 1 – Industrial and Automation Control Systems (IACS).

5.1.1 Metrics

5.1.1.1 Reducing Assurance and Certification Effort

{Q1} MC01.1 Automation of architecture-driven safety and security assurance process for the RTU

The system architecture will be modelled for assurance which allows a (semi)-automated safety and security assurance process. The effort of this approach will be compared to the current situation where no system architecture is modelled and which has no automation regarding safety and security assurance concerns.

{Q10} MC01.2 RTU compliance management effort

The number of activities and work products conceived in IEC 61508 and IEC 62443 and managed by the AMASS tools will be compared to a purely manual management flow. On the product level, cybersecurity certification schemes designed specifically for industrial control systems have a set of baseline security requirements that a product must satisfy. For example, the first commercial available certification of IACS products, the ISASecure EDSA certification scheme offers 3 levels to indicate the security level of a device. All three levels include certification elements of communication robustness testing, functional security assessment, and software development security assessment. In all 3 security levels, elements are specified by a list of product related security requirements. In addition, the artefacts of the certification schemes are mainly certification program documents, including technical specification, accreditation/recognition, symbol and certificates, and external references.

It should be noted that security assurance and security certification of IACS products have only developed in recent years, thus are rather new. For example, IEC 62443-4-2 “Security for industrial automation and control systems technical security requirements for IACS components” is just about to be officially published at the time of this writing. This is the industry’s approach to the need for security assurance while still makes the assurance effort practical and affordable comparing to previous assurance standard such as Common Criteria. Although it is not decided yet whether to certify a RTU against the ISASecure EDSA scheme within the project, it demonstrates the realistic requirements and efforts needed to prepare the certification.

In the context of ISASecure EDSA certification, clearly, efforts can be divided into two parts. The effort to conduct security testing by a third party, e.g. the accredited certifier in Europe TÜV; and the effort to manage assurance evidence and track specifications/requirements and testing results related to the product. Traditionally, this would be done by Word documents and Excel sheets. Thus, metric 2 is an estimation of the effort on managing text-based documents and model-based AMASS platform and its affiliated tools.

{Q2} MC01.3 Effort for determining the level of compliance of the RTU respect to the standards selected.

The reduction of manual work through the AMASS platform approach will reduce the effort to determinate the level of compliance of the RTU. In the context of certification scheme such as ISASecure ESDA, product level assurance means that the owner of a product must provide evidence that the product satisfies all specified requirements. The generation of the evidence is through analysis and practical testing. This also applies to safety assurance.

Safety & security co-analysis linked the security analysis with its physical impacts, which is appropriate for cyber-physical systems such as RTUs. Note that safety & security co-analysis is still under active research. No standard has been published that provides a common method for such a purpose. In AMASS project, FMVEA and Threat Modelling are used for safety & security analysis. The efficiency is manifested by

comparing the co-analysis method and its tool support for the RTU use case with traditional method such as FMVA. Note that the co-analysis provides results that are previous not available by traditional methods.

{Q1} MC01.4 Effort for running safety/security analysis of the RTU

Reduce time and error-proneness of the safety and security analysis process by means of automatic generation of, for example, FMVEA from system models.

5.1.1.2 Reusing Assurance Results

{Q7} MC01.5 Reuse of security and safety assurance results for other RTU platforms

Cost reduction regarding re-qualification/certification of previously qualified/certified assurance results will be quantified compared to totally manual activities. For instance, costs associated to the reuse. This could take place in the context of different RTUs.

As described in D4.2 [25], one important element of safety & security assurance case are the assurance arguments and assurance evidence organized in a structured way, e.g. GSN structure that is commonly recognized in the industry. It is suggested in D4.2 that assurance arguments and evidence can be clustered to form assurance patterns, which can be reused for other CPS components.

Current RTU and most IACS components from the same vendor use common software component. Therefore, it is possible that a component can obtain the status of safety and security assurance and such evidence can be reused for other products or platforms if the underlying software component is reused. For example, SSL is used in many RTU platforms for secure communication. Once analysed and tested against the relevant requirements, the “assurance pattern” on SSL component can be partially reused to provide support for the robustness and security of the SSL implementation in another platform, taking into consideration potential environment and configuration changes.

{Q7} MC01.6 Security and safety assurance reuse in RTU upgrade

In a similar manner to Metric 1, improvements in reuse costs could be measured for RTU upgrade. The assurance case by AMASS platform can provide the argument and evidence with tractability that can reduce the effort to make the assurance case from scratch.

{Q6} MC01.7 Reusing architecture-driven assurance results for RTUs

The effort in reusing architecture-driven assurance results such as V&V activities will significantly decrease. The AMASS approach will be compared to the current situation where no architecture-driven solutions are available.

5.1.1.3 Reducing Assurance and Certification and Qualification Risks

{Q11} MC01.8 Reduce the effort for identifying safety and security assurance risks for RTUs

In the scope of safety and security assurance of RTUs, risks associated to dependencies and trade-offs between safety and security concerns can arise. The identified and mitigated risks, and the discovered unknown risks related to safety and security assurance will be evaluated by using the AMASS platform.

Risk assessment and mitigation is transversal throughout the development lifecycle., e.g., requirements, design, implementation, and testing. Hazard Analysis and Risk Assessment (HARA) and Threat Analysis and Risk Assessment (TARA) go hand-in-hand to address safety and security risks. Standards such as SAE J3061 explicitly ask for HARA and TARA in the lifecycle for automotive systems. Co-analysis method in AMASS is the concrete step to conduct HARA and TARA in order to reduce risks. The metric can be calculated as the risk identified before HARA and TARA and using the co-analysis method.

{Q10} MC01.9 Reduce the effort for identifying architecture-based assurance risks for RTUs

The architecture-driven approach will help in reducing development risks associated to assurance and certification. This includes risks addressed by gaps associated between RTU components' information and RTU assurance needs. Safety & security co-analysis should be performed throughout development lifecycle, e.g. in requirement, design, and implementation phase. The report and result generated by co-analysis can be converted to different file formats to be integrated into tools related to RTU development. For example, a list of potential vulnerabilities and software weakness can be imported into a bug track system for design and test engineers.

{Q3} MC01.10 Reduce compliance management risks and automated documentation

Risk associated to safety and security compliance management will be measured comparing the current manual process to the one provided by AMASS. For both cases, the correctly identified, mitigated and newly identified risk will be measured with respect to the total ones.

{Q12} MC01.11 Reducing risks by seamless interoperability of tools related to RTU development

Through a seamless interoperability of AMASS tools, the rate of systematic failures will decrease. In order to quantify this metric, the process described in Section 3.3.3 "Reducing Risks by Seamless Interoperability" will be applied to RTUs.

5.1.1.4 Sustainable Impact by Harmonization and Interoperability***{Q16} MC01.12 Interoperability with other tools related to RTU development.***

The interoperability between tools will significantly reduce the manual work to be done, saving an amount effort in time and cost. The outcome of the security & safety co-analysis can be converted in different format, e.g. CSV file, which can be imported to other tools in the development process. For example, hazards and threats can be imported into existing requirement management systems, testing systems, or bug tracking systems to ensure they will be addressed in the development process.

5.1.2 Processes and Tools**5.1.2.1 Processes**

The processes for CS1 have been identified in D1.4 [26] section 3.1.

Processes for Usage Scenario 1 "Managing compliance with IEC 61508, IEC 62443 and IEC 62351":

- Standards Models Creation
- Assurance Project Creation
- Evidence Management
- Compliance Management

Processes for Usage Scenario 2 "Perform safety and security co-assessment":

- Model-based safety & Security product requirement management
- Safety & Security Co-analysis
- Safety & Security Assurance Case

5.1.2.2 Tools

The following tools will support the processes:

Tools support processes

Table 6. CS1 tools and processes

Process / Tools	OpenCert Tools: Standards Editor	OpenCert Tools: Assurance Project Management Editor	OpenCert Tools: Evidence Management Editor SVN repository to store actual evidence documents	OpenCert Tools: Assurance Project Management and Compliance Reporter Web Client	Model-based Requirement Management Tool (MORETO) Eclipse Papyrus	Excel sheet, optionally Microsoft Threat Modelling Tool for additional threat identification	MS Visio or assurance editor tool developed in AMASS
Standards Models Creation	X	-	-	-	-	-	-
Assurance Project Creation	-	X	-	-	-	-	-
Evidence Management	-	-	X	-	-	-	-
Compliance Management	-	-	-	X	-	-	-
Model-based safety & Security product requirement management	-	-	-	-	X	-	-
Safety & Security Co-analysis	-	-	-	-	-	X	-
Safety & Security Assurance Case	-	-	-	-	-	-	X

5.1.2.3 Prototype B

This iteration of the CS1 demonstrator will benchmark the following AMASS tool functionalities:

- Safety Integrity Level (SIL) / Security Level (SL) Accomplishment. It consists in assessing the level of compliance of the RTU project to achieve a given SIL and SL for certification.
- SIL/SL Gap Analysis. It provides a summary of compliance requirements to be met and evidence to be provided in order to achieve a given SIL and SL for certification.
- Specification of Safety-Security Interdependencies in the Assurance Case editors. It consists in specifying the dependencies of different safety and security assurance claims modelled in GSN by using the Assurance Case editor.

MORETO: SysML-based security modelling of RTU

- Models of the full IEC 62443-4-2 standard in requirement diagram
- Network architecture modelling in Block Definition Diagram (BDD)
- Component internal architecture in Internal Block Diagram (IBD)

Safety & security co-analysis tool

- Safety Failure Model analysis (in Excel Sheet)
- Threat modelling tool for threat and vulnerability analysis

Safety & security assurance case editor

- Patterns of assurance arguments and evidence for RTU

5.1.2.4 Prototype C

This iteration of the CS1 demonstrator will benchmark the following AMASS tool functionalities:

- Reuse of Assurance Projects (e.g. upgrade of the RTU version/model). The goal is to evaluate the extent to which AMASS reduces costs for reusing assurance and certification assets from one project into another representing a RTU upgrade (functionality or technology upgrade).
- Collaborative work of compliance management and assurance case edition. This aims at evaluating the AMASS facilities to work collaboratively: more than one user working in the AMASS tools and data at the same time.

MORETO

- Partial automation of requirements allocation
- Results importable to OpenCert

Safety & security co-analysis

- Interoperable results from failure mode analysis and threat modelling, which can also be imported to MORETO

Safety & security assurance case editor

- Results importable to OpenCert assurance case editor

5.2 Case Study 2

Case Study 2 – Advanced driver assistance function with electric vehicle sub-system (exemplified by one [CS2] or more [continued in CS3] model cars).

5.2.1 Metrics

5.2.1.1 Reducing Assurance and Certification Effort

The listed metrics herein will be used to measure a potential gain for design efficiency in development processes by reducing assurance and certification/qualification effort. This is especially relevant for a) product families and b) after replacing system components.

M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused

Using contracts we can reuse the assurance results for a subsystem in another context or system.

M6: Identification of architecture-based assurance risks

Using the AMASS platform and tools it is possible to identify automatically the risks by modelling the system.

M8: Addressing architecture-based assurance risks

The achievement regarding the metric M6 leads to a lower cost of assurance risks.

5.2.1.2 Reusing Assurance Results

The listed metrics herein will be used to measure a potential increase of reuse of assurance results (qualified or certified before), leading to cost reductions for component/product (re)certification/qualification activities. This is especially relevant for a) product families and b) after replacing system components.

M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused

Using contracts we can reuse the assurance results for a subsystem in another context or system.

M12: Assurance results reused across domains

The AMASS platform and methods provide the possibility to reuse the results across different domains (e.g. by means of contracts). Since Infineon is involved in both CS2 and CS7, the cross domain certification automotive – aviation will be considered.

5.2.1.3 Reducing Assurance and Certification and Qualification Risks

The listed metrics herein will be used to measure a potential reduction of assurance and certification/qualification risks of new safety-critical products.

M14: Identified risks related to architecture-driven assurance

The metric is calculated as the ratio of the risks automatically identified based on architecture-driven assurance, out of all the risks.

M16: Discovered unknown risks related to architecture-driven assurance

The metric is calculated as the ratio of the risks newly discovered based on architecture-driven assurance, out of all the risks.

5.2.1.4 Sustainable Impact by Harmonization and Interoperability

The listed metrics herein will be used to measure a potential increase of harmonization and interoperability of assurance and certification/qualification tool technologies.

M30: Common means for seamless interoperability

By means of the seamless interoperability of different tools within the AMASS platform, the manual work for transferring data from tool A to tool B will be reduced and the effort can be measured and compared.

5.2.2 Processes and Tools

5.2.2.1 Processes

Processes or activities that are to be measured are listed here:

- Definition of Project Development Cycle
- Compliance Management and Reporting of Compliance Result
- Safety and Dependability Assessment
- Verification & Validation

5.2.2.2 Tools

A number of software tools are used for static validation, such as PC-lint, AbsInt, Matlab/Simulink Verification & Validation toolbox.

5.3 Case Study 3

Case Study 3 – Cooperative ACC and Platooning (exemplified by a fleet of model cars).

5.3.1 Metrics

5.3.1.1 Reducing Assurance and Certification Effort

The listed metrics herein will be used to measure a potential gain for design efficiency in development processes by reducing assurance and certification/qualification effort.

M1: Automated architecture-driven and multi-concern assurance

With Tools like Savona and using SysML and contracts in comparison to conventional approaches, we can achieve a higher number of automated assurance objectives and hence an improvement of this metric.

M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused

Using contracts we can reuse the assurance results for a subsystem in another context or system.

M6: Identification of architecture-based assurance risks

Using the AMASS platform and tools it is possible to identify automatically the risks by modelling the system.

M8: Addressing architecture-based assurance risks

The achievement regarding the metric M6 leads to a lower cost of assurance risks.

5.3.1.2 Reusing Assurance Results

The listed metrics herein will be used to measure a potential increase of reuse of assurance results (qualified or certified before), leading to cost reductions for component/product (re)certification/qualification activities.

M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused

Using contracts we can reuse the assurance results for a subsystem in another context or system.

M12: Assurance results reused across domains

AMASS platform and methods provide the possibility to reuse the results across different domains (e.g. by means of contracts).

5.3.1.3 Reducing Assurance and Certification and Qualification Risks

The listed metrics herein will be used to measure a potential reduction of assurance and certification/qualification risks of new safety-critical products.

M14: Identified risks related to architecture-driven assurance, M15: Mitigated risks related to architecture-driven assurance and M16: Discovered unknown risks related to architecture-driven assurance.

A better performance regarding the metric M6 will lead to a better performance regarding M14. This will allow mitigating the identified risks for M15. Furthermore, using the methodical approaches of AMASS for the novel type of CPS we can discover the unknown risks related to the architecture-driven assurance.

5.3.1.4 Sustainable Impact by Harmonization and Interoperability

The listed metrics herein will be used to measure a potential increase of harmonization and interoperability of assurance and certification/qualification tool technologies.

M30: Common means for seamless interoperability

By means of the seamless interoperability of different tools within the AMASS platform, the manual work for transferring data from tool A to tool B will be reduced and the effort can be measured and compared.

5.3.2 Processes and Tools

5.3.2.1 Processes

Processes or activities that are to be measured are listed here:

- Definition of Project Development Cycle
- Compliance Management and Reporting of Compliance Result
- Safety and Dependability Assessment
- Verification & Validation

5.3.2.2 Tools

A number of software tools are used for static validation, such as PC-lint, AbsInt, Matlab/Simulink Verification & Validation toolbox.

Table 7. CS3 tools and processes

Process	DOORS	SAVONA under AMASS	TESTONA	Messina	EPF	APIS	Automated CFT under AMASS	Hansoft	Matlab Simulink/ Stateflow/ EmbeddedCoder	Rational Publishing Engine (RPE)	MS Word	MS Excel
Definition of Project Development Cycle	-	-	-	-	X	-	-	X	-	-	X	-
Natural Language - Requirement Specification	X	X	-	-	X	-	-	-	-	X	X	-
Semi-formal requirement specification/ Contract	-	X	-	-	-	-	-	-	-	-	-	-
Top-Down System Modelling	-	X	-	-	-	-	-	-	-	-	-	-
Controller Modelling	-	-	-	-	-	-	-	-	X	-	-	-
Safety Analysis	-	-	-	-	-	-	X	-	-	-	-	-
FMEA	-	-	-	-	-	X	-	-	-	-	-	-
Definition of Safety Mechanisms per Contracts	-	X	-	-	-	-	-	-	-	-	-	-

Safety and Dependability Assessment	X	-	-	-	-	-	-	-	X	-	-	-
Code Generation		-	-	-		-	-	-	X	-	-	-
Verification & Validation	X	X	X	X	X	-	X	X	X	X	X	X
Manual Test Case Specification	-	-	X	X	-	-	-	-	-	-	-	X
Automated Test Case Generation	-	-	X	-	-	-	-	-	-	-	-	-
HiL Test	-	-	-	X	-		-	-	-	-	-	-

5.4 Case Study 4

Case Study 4 – Design and safety assessment of on-board software applications in Space Systems.

During the development of the CS4, metrics will be taken for the new process and the new tools provided in the AMASS platform.

These new metrics will be compared with the metrics obtained in the original development of the OEU ICM SW, as the software was already developed and is being used.

5.4.1 Metrics

The Case Study 4 is based on three usages scenarios:

1. Component Reuse: assess the feasibility of components reuse using different execution platforms.
2. Re-qualification: analyse, at model level, the impact of a re-qualification when the HW platform is modified.
3. Safety analysis using the AMASS platform: analyse the system safety, performance, reliability and availability requirements using the AMASS platform.

So the metrics obtained in this use case shall be in line with the expected tasks. In this section a specific Goal-Question-Metric approach has been used in order to identify the metrics.

The main goals of CS4 are:

- Perform a system safety analysis using the AMASS platform
- Compare the effort needed for the safety analysis in the original project with the effort using AMASS platform
- Obtain the effort of performing again the safety analysis due to changes in the system specification

The CS4-specific metrics are listed below:

- Are RAMS issues discovered in early phases of the development?

{Q2} MC04.1 Number of issues discovered during design phases

Description: this metric calculates the ratio of RAMS issues found while designing software adding the RAMS aspects, out of the RAMS issues found during design phase when no adding RAMS aspects while designing software.

- In case of changes in the system specification, how many RAMS issues have changed?

{Q6} MC04.2 Ratio of RAMS issues that differ after a system specification change

Description: without automation the effort for obtaining all the RAMS issues have to be repeated in case of changes in the specification.

This metric aims to detect the effort that is not needed to repeat for the elements not affected by the specification change.

5.4.2 Processes and Tools

5.4.2.1 Processes

The Case Study 4 processes were defined in the document D1.2 [27].

5.4.2.2 Tools

The list of tools to be used in the CS4 is defined in the document D1.1 [24].

5.5 Case Study 5

Case Study 5 – Platform Screen Doors Controller.

There are two usage scenarios: US1 – generation of Frama-C asserted C code from B models and US2 – support for system-level model, including safety and security aspects. This is further described in D1.1 [24].

5.5.1 Metrics

M1. Automatic architecture-driven and multi-concern assurance

The system architecture will be modelled for assurance which allows a (semi)-automated safety and security assurance process. The effort of this approach will be compared to the current situation where no system architecture is modelled and which has no automation regarding safety concerns. Security assurance is out of the scope, as it will be performed during AMASS for the first time.

WP3 Metric 1. Percentage of requirements formalized

The requirements are at system-level (US2) and at code level (US1). This contributes directly to measure the capability for AMASS to fully support CS5 specification, for both US1 and US2.

WP3 Metric 4. Number of V&V activities automatically supported

The number of lines of code for which code review is replaced by proof.

$$\frac{\text{\# of code items proved}}{\text{total \# of code reviewed items}}$$

WP3 Metric 8. Percentage of requirements verified by V&V analysis

The percentage of automation for the code review, replaced by formal verification with Frama-C, from a property point of view.

$$\frac{\text{\# of automatically proved property}}{\text{total \# of property}}$$

WP4 Metric 4. Time needed for separate safety and security engineering process and the co-engineering process

Time required for the security analysis, in relation with the safety analysis. The effort is measured in man-minutes, man-hours, man-days, or other similar units, as appropriate.

{Q1} MC05.1 Effort spent on assurance activities

Each person working on a given development process will measure the time spent by him on this process. The measured time is accumulated for all the persons who perform the process. The effort is measured in man-minutes, man-hours, man-days, or other similar units, as appropriate.

{Q1} MC05.2 Security defaults detected

The number of defaults found during the modelling and analysis. No previous reference as the security analysis is performed for the first time during AMASS.

5.5.2 Processes and Tools

5.5.2.1 Process

- Capture requirements: collecting requirements for input specification
- Capture architecture: introducing the actual system-level architecture
- Formalize requirements: formalizing in Papyrus, in Atelier-B and in the asserted C code (generated) the requirements, at system and software level
- Define design: breaking down system level specification into sub-systems
- Perform formal verification: (automatically or interactively) proving or model checking verification elements
- Perform safety assessment: analysing the system from a safety point of view
- Perform security assessment: analysing the system from a security point of view

5.5.2.2 Tools

Table 8. CS5 tools and processes

Process	Atelier B	Frama-C	Papyrus
Capture requirements			x
Capture architecture			x
Allocate requirements			x
Formalize requirements	x	x	x
Define design			x
Perform formal verification	x	x	
Perform safety assessment			x
Perform security assessment			x

5.6 Case Study 6

CS6 – Automatic Train Control Formal Verification

The objective of the ALSTOM case study is to create a safety assurance project for an Automatic Train Control signalling system that includes formal proof demonstration for an Automatic Train Control railway signalling system (instead of classical workbench tests).

As per any Signalling Railway safety demonstration, several standards are generally applicable. In Europe and, usually for international project, the use of EN 50126, EN 50128 and EN 50129 are mandatory.

The safety assurance project shall include all artefacts required by the EN 50129 Generic Application Safety Case. These artefacts could be a reference to a document, table, diagram or text. The application of the EN 50129 requires independence between the designer, the verifier (V&V) and the safety validation team.

5.6.1 Metrics

{Q1} MC06.1 Cost of formal proof versus functional tests

Formally proving some safety properties on the system may replace some part of the verification and validation work. With this metrics, we want to estimate the cost of using formal methods. This is done with two ratios:

$$\frac{\text{\#of remaining tests} + \text{\#of Proof obligations}}{\text{total \# of tests}}$$

$$\frac{\text{effort needed to prove the obligations} + \text{effort needed for the remaining tests}}{\text{total effort needed for the tests}}$$

Our intuition is that using formal proof raises the validation cost (those ratios may be greater than one), but it decreases the cost of issue correction by detecting them earlier and raise the assurance.

Cost of formal proof

Number of proof obligations: This metric will count the number of generated proof obligations.

Effort needed to prove the obligations: This metric will count the estimated effort needed to prove the proof obligations. It is measured in man-hour, man-day or man-year.

Cost of functional tests

Number of remaining tests: This metrics will count the number of remaining functional tests.

Total number of tests: This metrics will count the number of functional tests.

Estimated effort needed for the remaining tests: This metric estimates the effort needed for the remaining tests. It is measured in man-hour, man-day or man-year.

Estimated effort needed for the tests: This metric estimates the effort needed for the tests. It is measured in man-hour, man-day or man-year.

{Q5} MC06.2 Early detection of safety issues

The decrease of cost of issue correction is measured by this ratio:

$$\frac{\text{Cost of early correction}}{\text{Estimated cost of late correction}}$$

Cost of early correction

For each issue, this metric will estimate the cost of correction of the issue. Since we are not allowed to share real costs, estimated cost will be normalised (for example: 0 for free correction and 1 for the maximal cost). The final metric will be the sum of all the cost for all the issues.

Estimated cost of late correction

For each issue, this metric estimates the cost of correction of the issue if the issue had been detected by classical means (*i.e.* functional tests). Once again, the cost will be normalised.

{Q1} MC06.3 Assurance raise thanks to use of the approach

This metric will estimate the assurance gap gained thanks to the use of formal methods. The safety validation process of a railway project is validated by an independent safety assessor (ISA). This assessor can do remarks on the process and on the safety evidence. The number of remarks by the ISA will be used to measure the assurance gap. It is measured by this ratio:

$$\frac{\text{\#of ISA remarks when using formal methods}}{\text{\#of ISA remarks when NOT using formal methods}}$$

Number of ISA remarks when using formal methods

This metric will count the number of ISA remarks on the project safety.

Number of ISA remarks when NOT using formal methods

This metric will count the number of ISA remarks on a similar project done without formal methods.

{Q3} MC06.4 Reducing qualification effort

For each system a set of feared events is defined through preliminary hazard analysis. For each feared event, a set of constraints on the system is defined to avoid the occurrence of the feared event. These constraints are written in the specification documents. To retrieve the constraints linked to a feared event, the feared event and the references to the documents that contain the corresponding constraints are generally stored in an Excel sheet. When one wants to check if a feared event is correctly covered, one must look for the correct Excel sheet, open it and search the constraints in the documents, which takes time (and may be source of error).

With the AMASS platform, the constraints could be directly linked to the event. With such a system, the access to the constraints associated to a feared event could be considered as instant in comparison. The number of constraints that can be stored in the platform will be used to measure the improvement. It is measured with the following ratio:

$$\frac{\text{\# of evidence stored in the platform}}{\text{total \#of evidence}}$$

Amount of evidence stored in the platform

This metrics will count the number of pieces of evidence that are stored in the AMASS platform.

Total amount of evidence

This metric will count the total number of pieces of evidence of the whole project.

{Q1} MC06.5 Automation of architecture driven assurance

The aim of this case study is to verify a safety property at system level, using formal methods. Starting from the general abstract safety property, more concrete properties will be derived. The architecture driven tools of the AMASS platform may help to automatically or semi automatically derive these properties. We are not sure yet about the applicability of those tools in the range of our case study, but it can be still be measured. This gain will be measured by a ratio:

$$\frac{\text{\# of automatically generated property}}{\text{total \# of property}}$$

Number of automatically generated property

This metric will count the number of properties that will be (semi-)automatically derived using the AMASS platform.

Number of property

This metric will count the total number of safety properties.

5.6.2 Processes and Tools

5.6.2.1 Process

- Proof Management Plan (to be included in Safety Plan): creating a document that sums up the action plan of the project.
- System Hazard Analysis (Informal Proof): this activity consists in
 - Identifying the safety properties by analysing the system specification, and;
 - explaining informally why the safety property is verified.
- Formal specification: this activity translates the informal specification into formal model.
- Formal Proof: this activity executes the formal proof with prover assistant tool.
- Formal model safety verification: this activity consists in verifying that the informal proof has correctly been translates into formal model.
- Process Assurance Evidence Recording: this activity consists in recording the results of the process for independent safety assurance and safety assessment.

5.6.2.2 Tools

Table 9. CS6 tools and processes

Process	Atelier B	Interactive Prover	Pro B	AMASS Platform	MS Word
Proof Management Plan	-	-	-	X	X
System Hazard Analysis (Informal Proof)	-	-	-	X	X
Formal Specification	X	-	-	-	
Proof	-	X	X	-	-
Formal model safety verification	-	-	-	X	X
Process Assurance Evidence Recording	-	-	-	X	-

5.7 Case Study 7

Case Study 7 – Safety Assessment of Multi-Modal Interactions in Cockpits.

The metrics will be measured during whole lifecycle for both the current process and the proposed improved process. The evaluation process introduced in iFEST [11] project will be reused and extended to cover Case Study 7 needs. For example, the Figure 6jError! No se encuentra el origen de la referencia. shows the evaluation process of the Cost of Poor Quality Improvement. The model allows simulation and execution so that it can be both verified for correctness and it can provide intermediate results.

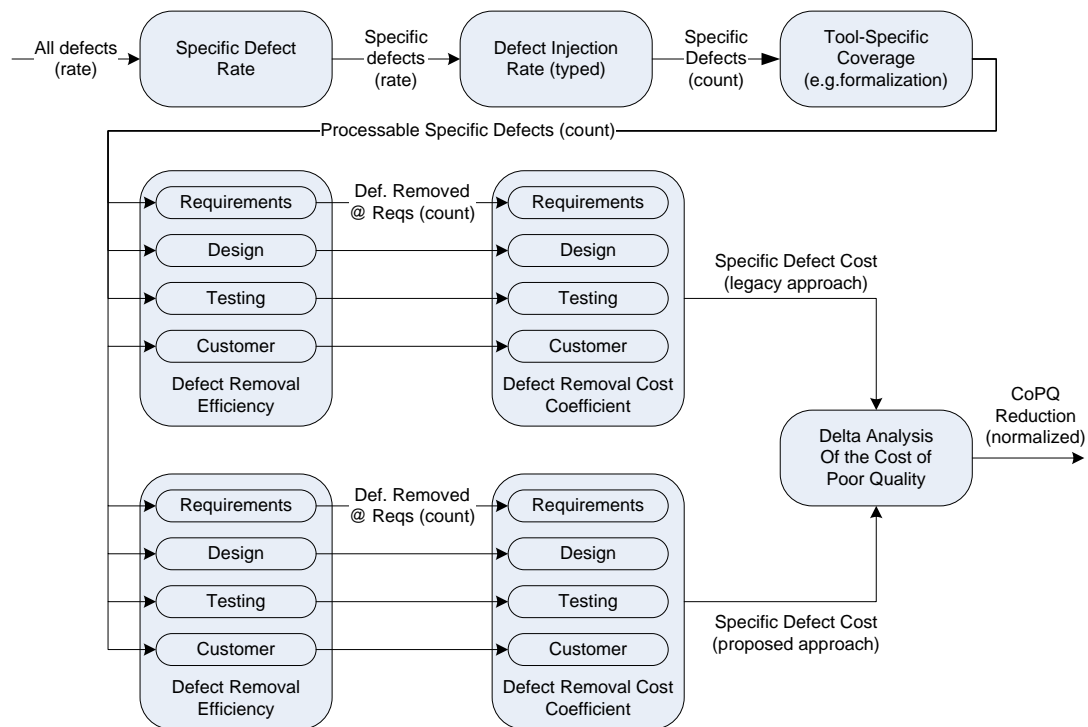


Figure 6. Evaluation process of CoPQ improvement

5.7.1 Metrics

5.7.1.1 Process-related metrics

{Q1} MC07.1 Effort Spent on Development Process

This metric contributes to the AMASS goal G1 and is actually a sub metric of common metric M1: Automated architecture-driven and multi-concern assurance. Each person working on a given development process will measure the time spent by him on this process. The measured time is accumulated for all the persons who perform the process.

The effort is measured in man-minutes, man-hours, man-days, or other similar units, as appropriate.

{Q10} MC07.2 Cost of Poor Quality of Development Process

This metric contributes to the AMASS goal G2 and sums up the metrics M14, M15, and M16.

The cost of poor quality (CoPQ) [6] are costs that would disappear if the processes, products, and tools were perfect.

There are two basic categories of cost of poor quality: direct and indirect.

The *direct poor-quality cost* is further subdivided into these categories: controllable poor-quality cost, resultant poor-quality cost, and equipment poor-quality cost. We will not be concerned with the *controllable poor-quality cost*, which is related to the quality planning, education and training, and conducting design reviews. The *equipment poor-quality cost* related to additional (not used during development) test equipment will not be measured, either. The *resultant poor-quality cost* has two components: the *internal error cost* which is not related to the customer, and the *external error cost* which consists of such items as sales returns, service level agreement penalties, complaint handling, and costs incurred due to warranty obligations.

The only CoPQ part partially observed in the CS7 will be the internal error cost. It consists of:

- In-process scrap and rework

- Troubleshooting and repairing
- Design changes
- Additional inventory required to support poor process yields and rejected lots (not observed)
- Re-inspection and retest of reworked items
- Downgrading (not observed)

The measurement will be performed similarly as the measurement of effort, also in the units like man-hours.

The *indirect poor-quality cost* (i.e. customer-incurred cost, customer-dissatisfaction cost, loss-of-reputation cost) will not be measured, since it is delayed and usually hard to quantify.

The cost of poor quality is measured in some monetary currency.

{Q5} MC07.3 Defect Introduced by Development Process

This metric contributes to the AMASS goal G1 and is actually a sub metric of common metric M8: Addressing architecture-based assurance risks. The number of defects is measured with regard to its source of occurrence (process).

{Q5} MC07.4 Defect Detected by Development Process

This metric contributes to the AMASS goal G1 and is actually a sub metric of common metric M8: Addressing architecture-based assurance. The number of defects is measured with regard to its point of detection (process).

{Q5} MC07.5 Defect Removed by Development Process

This metric contributes to the AMASS goal G1 and is actually a sub metric of common metric M8: Addressing architecture-based assurance risks. The number of defects that had to be removed while the given process is performed is measured. E.g. an error in the code has to be removed while the implementation process is being performed. More than one process can take part in removing one defect.

5.7.1.2 Product-related metrics

The product-related metrics count the number of a given kind of element contained in the project artefacts, e.g. the number of requirements, components, ports, etc.

5.7.2 Processes and Tools

5.7.2.1 Processes

The measured processes correspond to the processes identified in the D1.2 [27] in the CS7 section.

The following metrics will be measured on the processes:

- Effort spent on the process
- Cost of poor quality of the process
- # of defects introduced by the process
- # of defects detected by the process
- # of defects removed by the process

The description of the metrics is provided in the section 5.7.1.

The list of measured processes is here:

- Capture requirements
- Capture architecture

- Allocate requirements
- Formalize requirements
- Define design
- Perform formal verification
- Perform safety assessment
- Assess quality of evidence
- Store artefacts
- Generate reports
- Define development plan
- Define tasks and tools
- Report compliance results

5.7.2.1.1 Products

Requirements

- # of requirements
- # of formalized requirements

Architecture

- # of components
- # of ports
- # of flows

Design

- # of design elements (blocks, LOCs)

Verification results / Evidence

- # of passed verified requirements
- # of failed verified requirements

Report

Development plan

- # of processes
- # of product types

Tasks

- # of tasks of a given process

Tools

- # of tools

Standards

- # of standards

5.7.2.2 Tools

The following tools will support the processes:

Table 10. CS7 tools and processes

Process	V&V Manager	Property Manager	DiVinE	NuSMV	nuXmv	Matlab	EA	OCRA
Capture requirements	-	X	-	-	-	-	-	-
Capture architecture	-	-	-	-	-	X	X	-
Allocate requirements	-	X	-	-	-	-	X	-
Formalize requirements	-	X	-	-	-	-	-	-
Define design	-	-	-	-	-	X	-	-
Perform formal verification	X	-	X	X	X	-	-	-
Perform safety assessment	X	-	-	-	-	-	-	X
Assess quality of evidence	X	-	-	-	-	-	-	-
Store artefacts	-	-	-	-	-	-	-	-
Generate reports	X	-	-	-	-	-	-	-
Define development plan	-	-	-	-	-	-	-	-
Define tasks and tools	-	-	-	-	-	-	-	-
Report compliance results	-	-	-	-	-	-	-	-

5.8 Case Study 8

Case Study 8 – Telematics function.

The case study includes the specification, analysis, and assessment of an accurate positioning subsystem (element-out-of-context), where there are several quality attributes of vital importance, in particular safety and cybersecurity, to include in a multi-concern assurance case. There are three usage scenarios: US1 – creating a multi-concern assurance case, US2 – multi-concern assessment, and US3 – specification and analysis of safety, security and availability for an element out of context. This is further described in D1.1 [24].

5.8.1 Metrics

The metrics used in this case study are indicated in Table 5. There are no further metrics specific only to this case study.

5.8.2 Processes and Tools

5.8.2.1 Processes

The main process flows for the case study are described in more detail in D1.2 [27]. The main steps involved in metrics are:

- Assurance project creation (project initiation, standards tailoring, argumentation model)
- System design and dependability assessment (Requirements, design, dependability analysis)
- Evidence management
- Compliance management
- Assessment

For prototypes B and C these steps are also performed for a multi-concern assurance case.

5.8.2.2 Tools

Note that the tools mentioned below are those most relevant for the metrics. Additional tools not included in metrics collection are used for development, e.g. Eclipse, Git, KiCad. Some additional tool(s) may be added later e.g. for safety/security analysis.

5.8.2.2.1 Prototype A

Table 11. CS8 tools and processes, A

Process	Tools		
	MS Word	MS Excel	SVN
Project creation	X		
System design and dependability assessment	X	X	
Evidence management		X	X
Compliance management		X	
Assessment	X	X	X

5.8.2.2.2 Prototype B & C

Table 12. CS8 tools and processes, B & C

Process	Tools				
	OpenCert	EPF	Papyrus	MS Word	SVN
Project creation	X	X		X	
System design and dependability assessment			X	X	
Evidence management	X				X
Compliance management	X	X			
Assessment	X	X		X	

5.9 Case Study 9

Case Study 9: Safety-Critical SW Lifecycle of a Monitoring System for NavAid (ATM domain).

During the development of the case study, metrics will be evaluated for the new AMASS platform tools.

These results will be compared with the ones obtained in the current development tools.

5.9.1 Metrics

5.9.1.1 Reducing Assurance and Certification Effort

The following metrics will be used to evaluate the benefit in reducing the certification effort through the usage of AMASS tools.

The obtained results will be compared with the effort spent for previous certifications.

M2: Identification of consequences of CPS architecture on assurance and on certification/qualification

This metric will be used to evaluate how AMASS tools allow to automatically identify and generate architectural constraints that are needed to satisfy safety requirements.

M4: Architecture-driven assurance results and architecture-driven certification/qualification results reused

This metric will be used to evaluate if AMASS tools allow to reuse assurance results obtained in the development of the previous version of the same product.

M6: identification of architecture-based assurance risks

This metric will be used to evaluate how AMASS tools allow to automatically detect assurance risks, during the architecture definition.

M8: Addressing architecture-based assurance risks

This metric will be used to evaluate how the AMASS tools allow to reduce the costs for mitigating assurance risks.

5.9.1.2 Reducing Assurance and Certification and Qualification Risks

The following metrics will be used to evaluate the benefits in reducing the certification risks for safety critical product developments.

The obtained results will be compared with the risks that occurred during previous development and certification processes.

M14: Identified risks related to architecture-driven assurance

This metric will be used to evaluate how AMASS tools allow to automatically identify certification risks, during the architecture definition.

M15: Mitigated risks related to architecture-driven assurance

This metric will be used to evaluate how AMASS tools allow to mitigate the certification risks, during the architecture definition.

M16: Discovered unknown risks related to architecture-driven assurance

This metric will be used to evaluate how AMASS tools allow to automatically detect certification risks, during the architecture definition.

5.9.1.3 Metrics from WP3 – Architecture Driven Assurance***MW3.2 number of pieces of evidence and claims automatically generated (from contracts based design)***

This metric will be used to evaluate how AMASS tools allow to automatically generate evidences and claims. In current development process no evidences and claims are automatically generated.

MW3.4 number of V&V activities automatically supported

This metric will be used to evaluate how AMASS tools allow to automatically generate V&V activities. In current development process no V&V activities are automatically generated.

MW3.9 percentage of reduction of system design errors (automatically discovered by using contract-based design approach)

This metric will be used to evaluate how AMASS tools allow to reduce the system design errors. The system design errors present in the project, used as case study, will be compared with the system design errors in previous development projects (when available).

MW3.10 percentage of reduction of components integration errors (automatically discovered by using contract-based design approach)

This metric will be used to evaluate how AMASS tools allow to reduce the components integration errors. The components integration errors present in the project, used as case study, will be compared with the components integration errors in previous development projects (when available).

5.9.1.4 Metrics from the Case Study

{Q1} MC09.1 Effort spent on assurance activities

The total effort spent for assurance activities will be computed. Only the time needed for certification process itself will be considered, not the time for activities that would have been performed even if a certification process were not required. The certification process effort will be compared with the effort required in previous development projects (when available).

{Q2} MC09.2 Number of functional issues discovered during design phases

The AMASS tools should allow to early detect functional issues during the design phase, instead of later during V&V or integration phases. The total number of functional issues will be computed and compared with the ones detected during previous development projects (when available).

{Q10} MC09.3 Number of safety issues discovered during design phases

Same as MC09.2 but for safety issues.

5.9.2 Processes and Tools

5.9.2.1 Processes

The Case Study 9 processes were defined in the document D1.2 [27].

5.9.2.2 Tools

The list of tools to be used in the Case Study 9 is defined in the document D1.1 [24].

5.10 Case Study 10

Case study 10 - Certification basis to boost the usage of MPSoC architectures in the Space Market.

Metrics will be measured along the lifecycle of the project process. For each metric (or set of metrics) related to this space case study, information about how the metric could be enacted, the rationale for improvement, the metric measurement procedure, needs and constraints will be provided. Once the metrics have been further tailored and refined for the space domain, the current situation for each metric will be further specified.

5.10.1 Metrics

5.10.1.1 Code Metrics

A software metric is a standard of measure of a degree to which a software system or process possesses some property. The goal is to obtain objective, reproducible and quantifiable measurements, which may have numerous valuable applications in schedule and budget planning, cost estimation, quality assurance testing, software debugging, software performance optimization, and optimal personnel task assignments.

We can distinguish between two main code metrics for Space Domain:

{Q10} MC10.1 Estimate number of bugs in the code from static analysis and from dynamic execution of the code.

- From the static analysis, the McCabe's cyclomatic complexity hypothesis exposes that difficulty in understanding a program is largely determined by complexity of control flow graph.
- From dynamic execution, the main metric is to estimate the remaining number of bugs from the ones that have not been found yet. There are different failure count models (Binomial- type models, Non-

homogeneous Poisson Process Models (NHPP)... and Error seeding models (e.g. Halstead's software science model which estimates the number of errors in the program).

{Q10} MC10.2 Estimate the number of future failures.

- Input-Domain Models: Estimate program reliability using test cases sampled from input domain.
 - Partition input domain into equivalence classes, each of which usually associated with a program path.
 - Estimate conditional probability that a program is correct for all possible inputs, given that it is correct for a specified set of inputs.
 - Assume outcome of test case given information about behaviour for other points close to test point.
- Reliability Growth Models: Try to determine future time between failures.

Software Reliability: The probability that a program will perform its specified function for a stated time under specified conditions.

- Execute program until "failure" occurs, the underlying error found and removed (in zero time), and resume execution.
- Use a probability distribution function for the inter-failure time (assumed to be a random variable) to reduce future times to failure.
- Examining the nature of the sequence of elapsed times from one failure to the next.
- Assume occurrence of software failures is a stochastic process.

5.10.1.2 Programmer Productivity Metrics

Software development productivity is kind of intangible task (less code can be more productive than large code projects); it is not possible to measure directly.

If poor-quality software is produced quickly, it may appear to be more productive than if reliable and easy-to-maintain software is produced (measure only over software development phase).

- More does not always mean better.
- It may ultimately involve increased system maintenance costs.

Common measures to characterise the productivity:

{Q3} MC10.3 Lines of source code written per programmer month.

The number of lines of source code written by one programmer during one month.

{Q3} MC10.4 Object instructions produced per programmer month.

The number of object instructions produced by one programmer during one month.

{Q3} MC10.5 Pages of documentation written per programmer month.

The number of pages of documentation written by one programmer during one month.

{Q3} MC10.6 Test cases written and executed per programmer month.

The number of test cases written and executed by one programmer during one month.

5.10.1.3 Software Design Metrics

There are some general metrics to evaluate the quality of the software:

{Q3} MC10.7 Number of parameters

- Tries to capture coupling between modules.
- Understanding modules with large number of parameters will require more time and effort (assumption).
- Modifying modules with large number of parameters is likely to have side effects on other modules.

{Q3} MC10.8 Number of modules***{Q3} MC10.9 Number of modules called (estimating complexity of maintenance).***

- Fan-in: number of modules that call a particular module.
- Fan-out: how many other modules a module calls.

{Q3} MC10.10 Data Bindings: Triplet (p, X, q), where p and q are modules and X is a variable within scope of both p and q.

- Potential data binding:
 - X declared in both, but there is no indication of whether it is being accessed by a module.
 - Reflects possibility that p and q might communicate through the shared variable.

{Q3} MC10.11 Used data binding

- A potential data binding where p and q use X.
 - Harder to compute than potential data binding and requires more information about internal logic of module.

{Q3} MC10.12 Actual data binding

- Used data binding where p assigns value to x and q references it.
- Hardest to compute but indicates information flow from p to q.

{Q3} MC10.13 Cohesion metric

- Construct flow graph for module.
- Determine how many independent paths of the module go through the different statements.

5.10.1.4 Management Metrics***{Q18} MC10.14 Techniques for software cost estimation***

- Algorithmic cost modelling: Model developed using historical cost information that relates some software metric (usually lines of code) to project cost. Estimate is made based on the metric and then the model predicts the effort required.
- Expert judgement.
- Estimation by analogy: useful when other projects in same domain have been completed.
 - Parkinson's Law: Work expands to fill the time available.
 - Top-down estimation: cost estimate made by considering overall function and how functionality provided by interacting sub-functions. Made on basis of logical function rather than the components implementing that function.

5.10.2 Processes and Tools

The process has been defined and deployed to not only get the result of measurements, also history and trends. The main indicators to be monitored are:

- Monitoring of Project Status:
 - Total spent effort, effort to complete (for each lifecycle phase)
 - Evaluation of each lifecycle phase completion
 - Money used, cost to complete, estimated final cost
 - Percentage of delivered products (code, documents...)
- Monitoring of test phase:
 - Problems type distribution on the test lifecycle phases
 - Tests phases cost
 - Rate of detected, solved and remaining problems during test phase
 - Analysis of problem fixing during test phase.

The hierarchical organization of the process has adopted dedicated data acquisition procedures as explained in D1.2 [27]. For the process, the metrics explained above will be measured and studied. Deep inside the project some specific metrics will be treated.

In most cases the project starts with “Microsoft Project” to schedule the lifecycle of the project. A software project is a very complex process and its status is also a Multi-facet information.

The software development process is a complex network of different activities. The selected standard platforms make easy to interchange data between information systems, so no external equipment is used normally. Most of these activities suffer of some kind of re-work rate, so that when they are marked as “completed”, up to another 50% of the work is necessary to remove defects or adapt needed changes. Due to the different equipment used, technologies applied in each of them, the concrete technology for software development varies: LabView for mechanical proves with National Instruments boards, internal tool to integrate all boards with other or for scripting, “ISE Design Suite” for Xilinx FPGAs, Eclipse, Microsoft Visual Studio for programming software projects and to establish the software architecture. The rest of metrics/tools are summarised in the next table:

Table 13. CS10 tools and processes

Process	Doors	Jira	Melody	Microsoft Project	Microsoft Word/Sumo	Subversion
Capture requirements	X	-	-	-	X	-
Capture architecture	-	-	X	-	-	-
Formalize requirements	X	-	-	-	X	-
Define design	-	-	X	-	-	-
Perform formal verification	X	X	-	-	-	-
Software version management	-	-	-	-	-	X
Assess quality of evidence	X	-	-	-	-	-
Bugs reporting	-	X	-	-	-	-
Generate reports	-	X	-	-	-	-
Define development plan	-	-	-	X	-	-

Define tasks	-	X	-	-	-	-
Report compliance results	-	-	-	-	X	-

5.11 Case Study 11

Case Study 11 – Design and efficiency assessment of model based Attitude and Orbit Control software development.

5.11.1 Metrics

5.11.1.1 Reducing Assurance and Qualification Effort Metrics

The listed metrics herein will be used to measure a potential gain for design efficiency in AOCS SW development processes by reducing assurance and certification/qualification effort.

{Q1} MC11.1 Effort spent on assurance activities

The time spent on the assurance and qualification activities performing state of practice will be measured and compared to the time spent on the same activities performing state of the art using AMASS platform support.

{Q1} MC11.2 Rate of detected and solved issues during test phases

The rate of detected and solved issues performing state of practice will be measured and compared to the rate of detected and solved issues measured performing state of the art.

5.11.1.2 Reusing Assurance Result Metrics

The listed metrics herein will be used to measure a potential increase of reuse of assurance results (qualified or certified before), leading to cost reductions for component/product (re)certification/qualification activities.

{Q6} MC11.3 Reuse of contract based assurance

Through the use of contract based assurance measure the potential increase of reuse of safety and dependability assurance (FMECA).

5.11.1.3 Sustainable Impact by Harmonization and Interoperability Metrics

The listed metrics herein will be used to measure a potential increase of harmonization and interoperability of assurance and certification/qualification tool technologies.

{Q10} MC11.4 Manual work leading to poor quality

The reduction of manual work through the seamless integration of tools will be identified and the cost caused by manual errors will be measured and compared.

5.11.2 Processes and Tools

5.11.2.1 Processes

Processes or activities that are to be measured are listed here:

- Definition of Project Development Cycle
- Compliance Management and Reporting of Compliance Result
- Safety and Dependability Assessment

- Verification & Validation

5.11.2.2 Tools

A number of software tools are used for static validation, such as PC-lint, AbsInt, Matlab/Simulink Verification & Validation toolbox.

Table 14. CS11 tools and processes

Process	DOORS	EPF	Hansoft	Matlab/Simulink	Rational Publishing Engine (RPE)	MS Word
Definition of Project Development Cycle	-	X	X	-	-	-
Compliance Management and Reporting of Compliance Result	X	X	-	-	X	X
Safety and Dependability Assessment	X	-	-	X	-	-
Verification & Validation	X	-	X	X	X	X

6. Conclusions

The deliverable describes the AMASS evaluation framework, which will be used for assessment of the development process of the cases studies and of the contribution of the AMASS platform to the overall AMASS goals.

Metrics that have been derived from the AMASS goals using top-down approach (chapter 3). The corresponding measured values will provide the primary quantified assessment of how the AMASS goals are being fulfilled.

The work performed as part of the packages WP3, WP4, WP5, and WP6 will depend on or be supported by the measurement of the quantities (chapter 4). These quantities are originated from the identified needs and the advances made by the corresponding working groups.

Case-study-specific metrics are derived using a bottom-up approach and could measure a change in process or quality metrics beyond AMASS goals. These metrics are described in chapter 5. It was introduced in order to provide a better perception of the specific achievements made by the individual case studies.

The relation of the WP- and CS-specific metrics to the Questions of the GQM approach was captured by the references in curly brackets included immediately before each metric identifier, e.g. *{Q1} MC01.1 Metric name*.

Some metrics describe the properties of processes, while others are focused on the properties of artefacts.

The process-related metrics quantify e.g. the time consumed by a process, or the number of defects introduced or uncovered by a process. While the measurement of time can be less precise (e.g. due to omissions to start/stop the timewatch) and depend significantly on the measurement approach and in some cases on the willingness/discipline of the engineer, the defect related reports from a tracking system might be less subjective.

The artefact-related metrics will usually be generated by the tool that manages the artefacts. In order to automate the collection of such data, it is desirable to either adjust the metrics to the available functionality of the tools, or it could be necessary to develop a new plug-in of the AMASS tool platform that will automatically report the required metrics.

This deliverable will be revised in the future as the case studies are performed, including the specification of more detailed information where necessary.

Abbreviations and Definitions

AMASS	Architecture-driven, Multi-concern and Seamless Assurance and Certification of Cyber-Physical Systems
AOCS	Attitude and Orbit Control System
CACC	Cooperative Adaptive Cruise Control
CoPQ	Cost of Poor Quality
CPS	Cyber-Physical Systems
CS	Case Study
CSV	Comma-Separated Values
Dx.y	Deliverable, x .. WP number, y .. numeric identifier
EDSA	Embedded Device Security Assurance
EPF	Eclipse Process Framework
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
FMVEA	Failure Modes, Vulnerabilities and Effects Analysis
GQM	Goal-Question-Metric approach
GSN	Goal Structured Notation
Gx	Goal, x .. numeric identifier
HARA	Hazards Analysis & Risks Assessment
HIMSS	Healthcare Information and Management Systems Society
IACS	Industrial and Automation Control Systems
ICM	Instrument Control Module
IEC	International Electrotechnical Commission
ISA	Independent Safety Assessor
LISI	Levels Of Information Systems Interoperability
LOC	Line Of Code
MPSoC	Multiprocessor System-on-Chip
NHPP	Non-homogeneous Poisson Process Models
OEU	OLCI Electronics Unit
OLCI	Ocean & Land Colour Instrument
PrR	Product-related Reusability
RAMS	Reliability, Availability and Maintainability Analysis
RR	Relationship Ratio
RTU	Real Time Unit
SAE	Society of Automotive Engineers
SIL	Safety Integrity Level
SL	Security Level
SoC	System-on-Chip
SSL	Secure Sockets Layer
STO	Scientific and Technical Objective
SysML	Systems Modeling Language
TARA	Threat Assessment & Remediation Analysis
V&V	Verification and Validation
WP	Work Package

References

- [1] P. Koopman: Better Embedded System Software, Carnegie Melon University (2010)
- [2] RIA Proposal: AMASS (Technical Annex).
- [3] V. Basili, G. Caldiera, H.D. Rombach: Goal Question Metric Paradigm, Encyclopaedia of Software Engineering – 2 Volume Set (1994), <https://www.cs.umd.edu/~basili/publications/technical/T89.pdf>
- [4] Ebert, C., Dumke, R.: Software Measurement: Establish – Extract – Evaluate – Execute. Springer (2007)
- [5] N.E. Fenton, S.L. Pfleeger: Software Metrics – A Rigorous & Practical Approach, 2nd ed. PWS (1998)
- [6] https://en.wikipedia.org/wiki/Cost_of_poor_quality
- [7] [Christian Berger](#), [Holger Rendel](#), [Bernhard Rumpe](#): Measuring the Ability to Form a Product Line from Existing Products (2014), <https://arxiv.org/abs/1409.6583>
- [8] M. Kasunic: A Data Specification for Software Project Performance Measures: Results of a Collaboration on Performance Measurement (2008), http://resources.sei.cmu.edu/asset_files/TechnicalReport/2008_005_001_14987.pdf
- [9] R. van Solingen, E. Berghout: The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development, McGraw-Hill (1999)
- [10] A. Avizienis et al.: Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE (2004), https://www.nasa.gov/pdf/636745main_day_3-algirdas_avizienis.pdf
- [11] iFest - industrial Framework for Embedded Systems Tools, <http://www.artemis-ifest.eu/>
- [12] Crystal – Critical System Engineering Acceleration, http://cordis.europa.eu/project/rcn/111278_en.html
- [13] LISI Model: Levels Of Information Systems Interoperability (LISI) Reference Model <http://www.bmpcoe.org/library/books/lisi%20model/33.html>
- [14] ISA² - Interoperability solutions for public administrations, businesses and citizens, http://ec.europa.eu/isa/ready-to-use-solutions/imm_en.htm
- [15] Interoperability Maturity Mode <https://joinup.ec.europa.eu/elibrary/document/interoperability-maturity-model>
- [16] Interoperability requirements, TOGAF 9.1, <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap29.html>
- [17] Interoperability Levels for Dublin Core Metadata, <http://dublincore.org/documents/interoperability-levels/>
- [18] HIMSS interoperability standards, <http://www.himss.org/library/interoperability-standards/what-is-interoperability>
- [19] Understand the Three Levels of Interoperability, <http://blog.healthlanguage.com/understand-the-three-levels-of-interoperability>
- [20] Conceptual interoperability, https://en.wikipedia.org/wiki/Conceptual_interoperability#Levels_of_conceptual_interoperability
- [21] Design of the AMASS tools and methods for intra/cross domain reuse D6.2 (a)
- [22] Open Platform for Evolutionary Certification of Safety-critical Systems, OPENCROSS, <http://www.opencross-project.eu/>
- [23] Safety Certification of Software-Intensive Systems with Reusable Components, <http://www.safecer.eu/>
- [24] D1.1 Case studies description and business impact, November 2016
- [25] D4.2 Design of the AMASS tools and methods for multiconcern assurance (a)
- [26] D1.4 AMASS demonstrators (a), April 2017
- [27] D1.2 Report of case study data collection, March 2017

Appendix A Evaluation – EPF Process Description

This appendix summarises the Evaluation process in the form provided by the EPF Composer. This form is highly structured and optimized in the sense, that it instantiates an elaborated process meta-model, minimizes duplicated data, and generates several useful views of the same information in various contexts. The work breakdown structure is visualised in a type of activity/workflow diagram, which increases the understandability of the process description by providing a simple overview of the process components and their interdependence.

Evaluation

Delivery Process: Evaluation



The guidelines and a common framework that will be used by case studies providers and end user to assess the benefit/limitations of the AMASS solution.
The evaluation framework consists of the set of metrics for assessing the achievements of AMASS goals.

Description

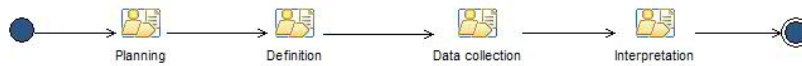
Work Breakdown Structure

Team Allocation

Work Product Usage

[Expand All Sections](#)
[Collapse All Sections](#)

Workflow


[Back to top](#)

Work Breakdown

[Expand All Sections](#)
[Collapse All Sections](#)

Breakdown Element	Steps	Index	Predecessors	Model Info	Type	Planned	Repeatable	Multiple Occurrences	Ongoing	Event Driven	Optional	Team
Planning		1			Capability Pattern	✓						
Definition		5	1		Capability Pattern	✓						
Data collection		7	5		Capability Pattern	✓						
Interpretation		10	7		Capability Pattern	✓						

Figure 7. Top-level overview of the evaluation process

Evaluation > Planning

Capability Pattern: Planning

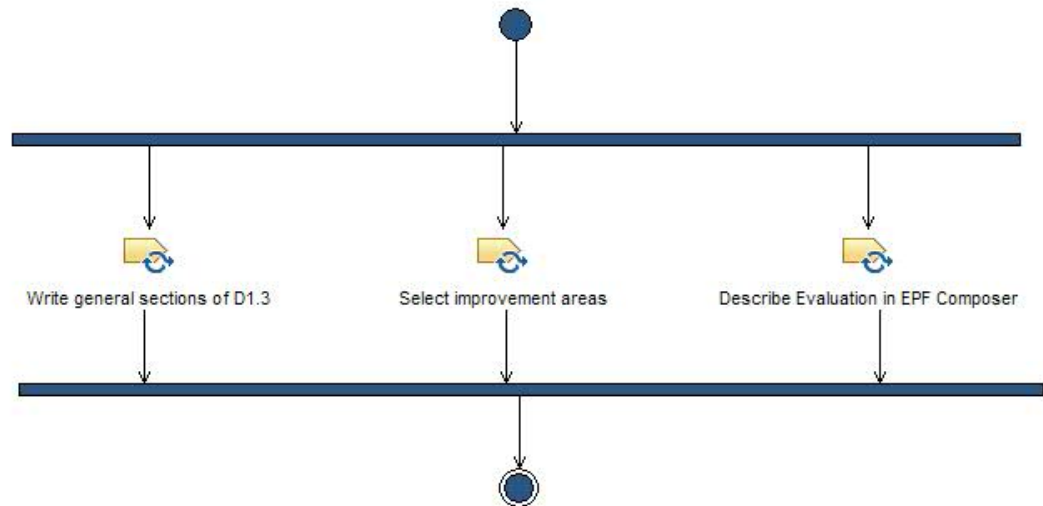

Extends: Planning

Description

Work Breakdown Structure

Team Allocation

Work Product Usage

Workflow

Work Breakdown

Breakdown Element	Steps	Index	Predecessors	Model Info	Type	Planned	Repeatable	Multip
Write general sections of D1.3		2			Task Descriptor			
Select improvement areas		3			Task Descriptor			
Describe Evaluation in EPF Composer		4			Task Descriptor			


Figure 8. Activity diagram of the planning process

Task: Write general sections of D1.3

Write the following sections of the deliverable D1.3 Evaluation Framework and Quality Metrics:

- Executive Summary
- 1. Introduction
- 2. Common Evaluation Foundation
- 2.1 Goal-Question-Metric Approach
- 2.4 Common Evaluation Framework
- 3. AMASS Metrics
- 3.5 Common Evaluation Procedures
- 3.6 Common Evaluation Framework
- 4. Technical Solution Metrics, Processes, and Tools
- Conclusions

Disciplines: [Planning](#)

 Expand

Relationships		
Roles	Primary Performer: <ul style="list-style-type: none"> • GQM coach • GQM manager 	Additional Performers:
Outputs	<ul style="list-style-type: none"> • D1.3 Evaluation Framework and Quality Metrics 	

Figure 9. Description of the task *Write general sections of D1.3*

Task: Select improvement areas

Select suitable product or process improvement ideas.

Write this section of D1.3 document:

- 2.2 AMASS Goals and Objectives.

Disciplines: [Planning](#)

Purpose	
Top level of the question: Are we measuring the right thing?	
Relationships	
Outputs	<ul style="list-style-type: none"> • D1.3 Evaluation Framework and Quality Metrics

Figure 10. Description of the task *Select improvement areas*

Task: Describe Evaluation in EPF Composer



Create a process description of the evaluation of the (AMASS) solution in the EPF composer.

Disciplines: [Planning](#)

[Expand All Sections](#)

[Collapse All Sections](#)

▢ Purpose

Are we measuring the key things right?

[Back to top](#)

▢ Relationships

Roles	Primary Performer: <ul style="list-style-type: none"> GQM coach GQM support engineer 	Additional Performers: <ul style="list-style-type: none">
Inputs	Mandatory: <ul style="list-style-type: none"> D1.3 Evaluation Framework and Quality Metrics 	Optional: <ul style="list-style-type: none"> None
Outputs	<ul style="list-style-type: none"> EPF process description 	

[Back to top](#)

▢ More Information

Supporting Materials	<ul style="list-style-type: none"> Performance measurement The Goal Question Metric Method
-----------------------------	--

Figure 11. Description of the task *Describe Evaluation in EPF Composer*

Evaluation > Definition

Capability Pattern: Definition



Extends: Definition

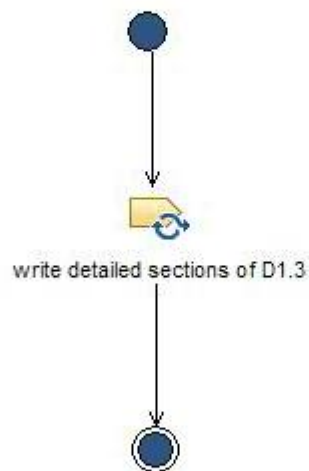
Description

Work Breakdown Structure

Team Allocation

Work Product Usage

Workflow



Work Breakdown

Breakdown Element	Steps	Index	Predecessors	Model Info	Type	Planned
write detailed sections of D1.3		6			Task Descriptor	

Figure 12. The Definition phase of the evaluation consists of just one major task

Task: write detailed sections of D1.3

Write the following sections of the deliverable D1.3 Evaluation Framework and Quality Metrics:

- 2.3 Software Related Metrics
- 3.1 Reducing Assurance and Certification Effort
- 3.2 Reusing Assurance Results
- 3.3 Reducing Assurance and Certification and Qualification Risks
- 3.4 Sustainable Impact by Harmonization and Interoperability
- 4.1 Metrics from WP3 – Architecture Driven Assurance
- 4.2 Metrics from WP4 – Multi-Concern Assurance
- 4.3 Metrics from WP5 – Seamless Interoperability
- 4.4 Metrics from WP6 – Cross/Intra Domain Reuse
- 5. ... x. Case Study 1..11 Specific Metrics, Processes, and Tools

Disciplines: [Definition](#)

[Expand All Sections](#)
[Collapse All Sections](#)
Relationships

Inputs	Mandatory: <ul style="list-style-type: none"> • D1.3 Evaluation Framework and Quality Metrics 	Optional: <ul style="list-style-type: none"> • None
Outputs	<ul style="list-style-type: none"> • D1.3 Evaluation Framework and Quality Metrics 	

[Back to top](#)
Main Description

The subjects of measurements are either *processes* or *things*.

The basic quantity measured on **processes** is the *effort*. For effort-related metrics it is important to clarify which activities shall be observed.

The basic quantity measured on **things** is a whole *number* / a count of some structural units/items. For number/count-related metrics the specification of counted items needs to be correct, consistent, unambiguous, complete, otherwise the to be counted items might not be properly identified and the results of the measurements would have low repeatability.

Example:

Open assurance and certification tool users .. Number of tool users. This metric expresses the number of individuals, who perform assurance/certification related activities and while performing them they use one or more tools of the AMASS platform.

Figure 13. The description of the task *Write detailed sections of D1.3*

Evaluation > Data collection

Capability Pattern: Data collection



Extends: Data collection

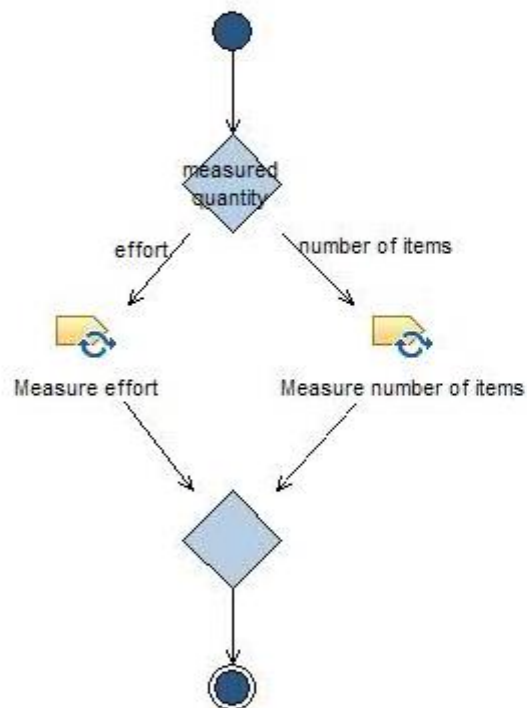
Description

Work Breakdown Structure

Team Allocation

Work Product Usage

Workflow



Work Breakdown

Breakdown Element	Steps	Index	Predecessors	Model Info	Type	Planned F
Measure effort		8			Task Descriptor	
Measure number of items		9			Task Descriptor	

Figure 14. Activity diagram for the *Data collection* phase

Task: Measure effort


Disciplines: [Data collection](#)
[Expand All Sections](#)
[Collapse All Sections](#)

Relationships

Roles	Primary Performer: <ul style="list-style-type: none"> Project Manager Quality Assurance SW Developer Tester 	Additional Performers: <ul style="list-style-type: none"> GQM coach
Inputs	Mandatory: <ul style="list-style-type: none"> D1.3 Evaluation Framework and Quality Metrics 	Optional: <ul style="list-style-type: none"> None
Outputs	<ul style="list-style-type: none"> D1.7 Case study implementation and benchmarking measured data 	

[Back to top](#)

Main Description

Effort is the total team time that is spent on selected project-related activities during the life cycle of a project. Activities that do not specifically contribute to the development and delivery of the software products are excluded from the calculation of effort.

The total team time is the sum of times spent on the selected activities by individual team members.

[Effort template](#)

[Back to top](#)

Key Considerations

Clarify which activities shall be observed.

Make all team members whose work shall be included into the measurement aware of this fact.

[Back to top](#)

More Information

Supporting Materials	<ul style="list-style-type: none"> Performance measurement
Tool Mentors	<ul style="list-style-type: none"> Measurement support system

Figure 15. The description of the task *Measure effort*

Template: Effort template

[Expand All Sections](#)
[Collapse All Sections](#)
Main Description
Example

A project team of 6 individuals recorded their time spent on project-related activities and reported the information at the end of each week. When the project was completed, the cumulative hours for each team member were calculated and the following table was produced.

Id	Team member	Hours
1	Project Manager	50
2	Requirements Analyst	20
3	Software Developer	120
4	Software Developer	30
5	Software Tester	30
6	Quality Assurance	10
	TOTAL	260

Figure 16. An example of the table with the recorded effort

Task: Measure number of items


Disciplines: [Data collection](#)
[Expand All Sections](#)
[Collapse All Sections](#)

Relationships

Inputs	Mandatory: <ul style="list-style-type: none"> D1.3 Evaluation Framework and Quality Metrics 	Optional: <ul style="list-style-type: none"> None
Outputs	<ul style="list-style-type: none"> D1.7 Case study implementation and benchmarking measured data 	

[Back to top](#)

Main Description

Find all instances that meet the specification and count them.

[Back to top](#)

Key Considerations

The specification of counted items is correct, consistent, non-ambiguous, complete.

Sufficient clarity to all who will use the specification for counting the items is needed: they have to know where to find the items and how to filter them.

[Back to top](#)

More Information

Tool Mentors	<ul style="list-style-type: none"> Measurement support system
---------------------	--

Figure 17. The description of the task *Measure number of items*

Evaluation > Interpretation

Capability Pattern: Interpretation



By using the collected data, answer the questions underlying the GQM based evaluation.

Extends: Interpretation

Description

Work Breakdown Structure

Team Allocation

Work Product Usage

Workflow



Work Breakdown

Breakdown Element	Steps	Index	Predecessors	Model Info	Type
Analyze collected data and answer questions		11			Task Descriptor


Figure 18. The overview of the Interpretation phase

Task: Analyze collected data and answer questions




Answer the questions as defined in the GQM plan, and based on these answers, one should be able to conclude whether the defined measurement goals are attained.

 Expand All Sections

 Collapse All Sections

Relationships

Roles	Primary Performer: <ul style="list-style-type: none"> GQM coach GQM manager 	Additional Performers: <ul style="list-style-type: none"> Project Manager Quality Assurance SW Developer Tester
Inputs	Mandatory: <ul style="list-style-type: none"> D1.7 Case study implementation and benchmarking measured data 	Optional: <ul style="list-style-type: none"> None
Outputs	<ul style="list-style-type: none"> D1.7 Case study implementation and benchmarking 	

 [Back to top](#)

More Information

Tool Mentors	<ul style="list-style-type: none"> Measurement support system
---------------------	--

Figure 19. The description of the task *Analyze collected data and answer questions*

Description	Work Breakdown Structure	Team Allocation	Work Product Usage
Team Breakdown			
Breakdown Element	Model Info	Team	Type
GQM coach			Role Descriptor
D1.3 Evaluation Framework and Quality Metrics	Modifies		Deliverable Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
EPF process description	Modifies		Artifact Descriptor
Analyze collected data and answer questions	Performs as Owner		Task Descriptor
Describe Evaluation in EPF Composer	Performs as Owner		Task Descriptor
Write general sections of D1.3	Performs as Owner		Task Descriptor
Measure effort	Performs as Additional		Task Descriptor
GQM manager			Role Descriptor
D1.3 Evaluation Framework and Quality Metrics	Modifies		Deliverable Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
Analyze collected data and answer questions	Performs as Owner		Task Descriptor
Write general sections of D1.3	Performs as Owner		Task Descriptor
GQM support engineer			Role Descriptor
EPF process description	Modifies		Artifact Descriptor
Describe Evaluation in EPF Composer	Performs as Owner		Task Descriptor

Figure 20. The roles that support the Goal-Question-Metric approach

Description	Work Breakdown Structure	Team Allocation	Work Product Usage
<div> <div></div> <div>Team Breakdown</div> <div> <div></div> <div>Expand</div> </div> </div>			
Breakdown Element	Model Info	Team	Type
<div></div> GQM coach			Role Descriptor
<div></div> GQM manager			Role Descriptor
<div></div> GQM support engineer			Role Descriptor
<div></div> Project Manager			Role Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
measured data	Modifies		Artifact Descriptor
Measure effort	Performs as Owner		Task Descriptor
Analyze collected data and answer questions	Performs as Additional		Task Descriptor
<div></div> Quality Assurance			Role Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
measured data	Modifies		Artifact Descriptor
Measure effort	Performs as Owner		Task Descriptor
Analyze collected data and answer questions	Performs as Additional		Task Descriptor
<div></div> SW Developer			Role Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
measured data	Modifies		Artifact Descriptor
Measure effort	Performs as Owner		Task Descriptor
Analyze collected data and answer questions	Performs as Additional		Task Descriptor
<div></div> Tester			Role Descriptor
D1.7 Case study implementation and benchmarking	Modifies		Deliverable Descriptor
measured data	Modifies		Artifact Descriptor
Measure effort	Performs as Owner		Task Descriptor
Analyze collected data and answer questions	Performs as Additional		Task Descriptor

Figure 21. The roles of the development team

Deliverable: D1.3 Evaluation Framework and Quality Metrics


AMASS_D1.3_WP1.docx

Expand All Sections

Collapse All Sections

Relationships		
Roles	Responsible:	Modified By: <ul style="list-style-type: none"> • GQM coach • GQM manager
Tasks	Input To: <ul style="list-style-type: none"> • Describe Evaluation in EPF Composer • Measure effort • Measure number of items • write detailed sections of D1.3 	Output From: <ul style="list-style-type: none"> • Select improvement areas • write detailed sections of D1.3 • Write general sections of D1.3

Back to top

Main Description

See https://services-medini.kpit.com/svn/AMASS_collab/WP1/D1.3_in_progress/AMASS_D1.3_WP1.docx .

Figure 22. The description of the deliverable *D1.3 Evaluation Framework and Quality Metrics*

Deliverable: D1.7 Case study implementation and benchmarking


Expand All Sections

Collapse All Sections

Relationships		
Roles	Responsible:	Modified By: <ul style="list-style-type: none"> • GQM coach • GQM manager • Project Manager • Quality Assurance • SW Developer • Tester
Tasks	Input To: <ul style="list-style-type: none"> • Analyze collected data and answer questions 	Output From: <ul style="list-style-type: none"> • Analyze collected data and answer questions • Measure effort • Measure number of items

Figure 23. The description of the deliverable *D1.7 Case study implementation and benchmarking*

Artifact: EPF process description


This EPF model.

Expand All Sections

Collapse All Sections

Relationships		
Roles	Responsible:	Modified By: <ul style="list-style-type: none"> • GQM coach • GQM support engineer
Tasks	Input To:	Output From: <ul style="list-style-type: none"> • Describe Evaluation in EPF Composer

Figure 24. The process description model

Artifact: measured data

[Expand All Sections](#) [Collapse All Sections](#)

Relationships		
Roles	Responsible:	Modified By: <ul style="list-style-type: none"> Project Manager Quality Assurance SW Developer Tester
Tasks	Input To: <ul style="list-style-type: none"> Analyze collected data and answer questions 	Output From: <ul style="list-style-type: none"> Measure effort Measure number of items

Figure 25. The measured data artifact