vf-OS: virtual factory Operating System

WP8: vf Smart Application Piloting and Validation

D8.1c: Validation Scenarios - Vs: 1.0.3

Deliverable Lead and Editor: Raúl Poler, UPV

Contributing Partners: UPV, MASS, VS, IKERLAN, KBZ, CON, APR, TARDY, UNINOVA

Date: 2018-10

Dissemination: Public

Status: Consortium Approved

Short Abstract

This document serves as umbrella document for the evaluation of the vf-OS project pilots. It presents a revision of the roadmap and supporting documents for evaluators.

Grant Agreement: 723710

Document Status

Deliverable Lead	Raúl Poler, UPV
Internal Reviewer 1	Carlos Agostinho, KBZ
Internal Reviewer 2	Eduardo Saiz, IKERLAN
Internal Reviewer 3	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP8: vf Smart Application Piloting and Validation
ID	D8.1c: Validation Scenarios
Due Date	2018-09
Delivery Date	2018-10
Status	Consortium Approved

History

See Annex A.

Status

This deliverable is subject to final acceptance by the European Commission.

Further Information

www.vf-OS.eu

Disclaimer

The views represented in this document only reflect the views of the authors and not the views of the European Union. The European Union is not liable for any use that may be made of the information contained in this document.

Furthermore, the information is provided “as is” and no guarantee or warranty is given that the information is fit for any particular purpose. The user of the information uses it at its sole risk and liability.

Project Partners:

ICE Information Catalyst

UNIVERSITÉ LUMIÈRE LYON 2

IK4 IKERLAN Research Alliance

MONDRAGON ASSEMBLY

UNINOVA Instituto de Desenvolvimento de Novas Tecnologias

VIASOLIS

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

CONSULGAL

Caixa Mágica Software

KNOWLEDGE BIZ CONSULTING

almende ORGANIZING NETWORKS

APR Applications Plastiques du Rhône

ASCORA Consulting - Development - Research

TARDY

Executive Summary

Deliverable “D8.1c Validation Scenarios” and its annexed documents describe the evaluation and validation activities that will be conducted using the pilot applications (vApps) defined in the scope of the different validation scenarios. This document provides a revision of the evaluation roadmap, the main objectives of the evaluation activities, the dependencies between vf-OS Platform components and vApps, and the means to verify that the evaluation covers all the functionalities of the vf-OS Platform. It also includes a description of the role of every partner in the different use cases, and guidelines to conduct the evaluation.

Each of the annexes provides an evaluation report of the validation activities per pilot:

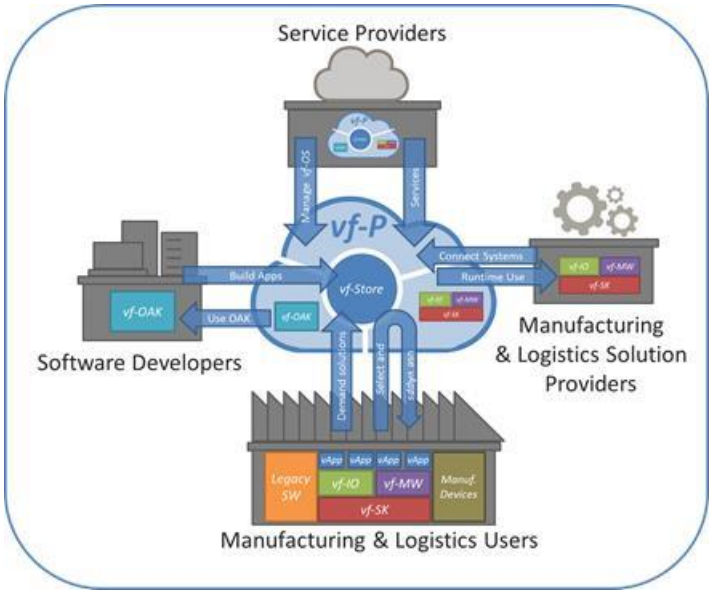
- **Pilot 1: Manufacturing and Logistics / Automation:** Applications focused on providing support to maintenance and spare part stock management in automation production equipment. At M24, the evaluation activities in Pilot 1 are focused on the interoperability between the vf-OS Platform and the production equipment
- **Pilot 2: Construction / Industrialisation:** Applications to improve time and resource utilisation in construction projects through advanced collaboration between construction managers and material suppliers. At M24, the evaluation activities in Pilot 2 address Enablers (both FI-WARE Enablers and vf-OS Enablers)
- **Pilot 3: Manufacturing Assembly / Collaboration:** Applications to enable collaborative design and manufacturing of complex products. At M24, the evaluation activities in this pilot are focused on the interoperability between the vf-OS Platform and legacy software systems in distributed environments

Table of Contents

0 Introduction 1
1 Evaluation Roadmap 5
2 Mapping of Development and Evaluation Activities 7
2.1 Pilot 1 Mapping 8
2.2 Pilot 2 Mapping 9
2.3 Pilot 3 Mapping 10
2.4 Completeness Check 12
3 Evaluation Roles and Guidelines 15
3.1 Evaluation Roles 15
3.2 Evaluation Guidelines 16
4 Conclusions 21

0 Introduction

0.1 vf-OS Project Overview



vf-OS – virtual factory Open Operating System – is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 723710 and conducted in the period October 2016 until September 2019. It engages 14 partners (Users, Technology Providers, Consultants and Research Institutes) from 7 countries with a total budget of circa 7.5M€. Further information can be found at www.vf-OS.eu.

The World is facing the fourth industrial revolution based on ICT, specifically architectures and services, as key innovation drivers for manufacturing companies. Traditional factories will increasingly be transformed into smart digital manufacturing environments but currently the full potential for ICT in manufacturing is far from being fully exploited. Factories are complex systems of systems and there is a need to develop a platform on which future manufacturing applications can be built. Examples of platforms exist in some industrial sectors but there is a lack of cross cutting platforms based on open standards for creating an ecosystem for cooperative innovation. Innovative open platforms to attract talent from solution developers and to provide accessible manufacturing smart applications to European SMEs are examples of the kind of solutions being sought.

The goal of vf-OS is to develop an Open Operating System for Virtual Factories composed of a kernel, application programming interface, and middleware specifically designed for the factory of the future. An Open Applications Development Kit (OAK) will be provided to software developers for deploying Manufacturing Smart Applications for industrial users, using the vf-OS Manufacturing Applications Store all operated through a Virtual Factory Platform.

The Virtual Factory Platform is an economical multi-sided market platform with the aim of creating value by enabling interactions between four customer groups:

- **Software Developers (independent or within individual manufacturers)** which will build Manufacturing Apps either through innovation or from manufacturing user demand
- **Manufacturing and Logistic Users** which will explore the marketplace for already created solutions, ready to be run on the vf-OS
- **Manufacturing and Logistics Solutions Providers** which will provide ICT interfaces and manufacturing connections

- **Service Providers (vf-OS innovators and third parties)** will make available services (hosting, storage, connected cloud services, etc.) including those based on developed solutions

The Virtual Factory Platform will provide a range of services to the connected factory of the future to integrate better manufacturing and logistics processes. Manufacturing Applications Store will be open to software developers who, using the free Open Applications Development Kit provided, will be able to quickly develop and deploy smart applications to enable, and optimise communication and collaboration among supply networks of all manufacturing sectors in all the manufacturing stages and logistic processes.

vf-OS aims to become the reference system software for managing factory related computer hardware and software resources and providing common services for factory computational programs. This operating system will be the component of the system software in a real factory system where all factory application programs will run.

0.2 Deliverable Purpose and Scope

Deliverable D8.1c is a report detailing the validation activities using vApps requested in the vf-OS pilot domains, as agreed between the vf-OS Software Developers and Manufacturing Logistic Users involved in each project pilot. This deliverable describes:

- The objectives of the evaluation activities on every evaluation phase
- The dependencies between developed components and vApps functionalities, and an aggregated plan of vApp releases regarding vf-OS component availability that is detailed in the annexes
- The pilot applications evaluated in every phase
- The roles of each partner in the pilot and evaluation guidelines for all pilots.

This document acts as an umbrella document for the reports providing the results of the evaluation of pilot applications at different stages of the project, following the guidelines defined in this document. The evaluation reports are presented in the annexes C, D and E of this document, corresponding to each pilot.

0.3 Target Audience

This public deliverable describes the vApps functionalities used to validate the vf-OS project and therefore, it is primarily aimed at project partners. However, since the vApps can be used to solve common problems to different sectors, the document can be useful for the wider scientific and industrial community. This includes any other organisation that may be interested in the potential functionality of the vf-OS technology, as well as other publicly funded projects which may be interested in collaboration activities.

0.4 Deliverable Context

This document is one of the main documents supporting the development of the applications that will be used to validate the vf-OS project. Its relationship to other documents is as follows:

- **Vision Consensus (D1.1):** A report to strengthen the common understanding of the project vision and the overall aims and objectives of the project. Available now
- **User Scenarios Characterisation (D1.2):** A report providing the characterisation of the main industrial scenarios and the initial pilot scenarios description. Available now
- **Requirements Specifications (D1.5):** The requirements specifications for the vf-OS Platform and the pilot vApps. Available now
- **Validation Scenarios (D8.1a):** Provides the methodology that establishes a reference for the interactions between different vf-OS customer groups developing and using new vApps. Available now
- **Validation Scenarios (D8.1b):** Defines the functionalities of the different pilot applications (vApps) in the project pilots. Available now
- **Pilot 1: Manufacturing & Logistic – Automation (D8.2abc):** A series of demonstrators of the applications developed for pilot domain 1, reported in Annex C of this deliverable
- **Pilot 2: Construction – Industrialisation (D8.3abc):** A series of demonstrators of the applications developed for pilot domain 2, reported in Annex D of this deliverable
- **Pilot 3: Manufacturing Assembly – Collaboration (D8.4abc):** A series of demonstrators of the applications developed for pilot domain 3, reported in Annex E of this deliverable

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section1:** **Evaluation Roadmap:** A revision of the evaluation objectives and roadmap
- **Section2:** **Mapping of Development and Evaluation Activities:** A description of the dependencies between pilot vApps and platform components, accompanied by the calendar of components tests for every pilot
- **Section3:** **Evaluation Roles and Guidelines:** The roles assigned to each partner in each pilot as defined in the validation methodology

- **Annexes**:
  - **Annex A**: Document History
  - **Annex B**: References
  - **Annex C:** Pilot 1 Evaluation Report (Annexed Document)
  - **Annex D:** Pilot 2 Evaluation Report (Annexed Document)
  - **Annex E:** Pilot 3 Evaluation Report (Annexed Document)

0.6 Document Status

This document is listed in the Description of Action as “public”.

0.7 Document Dependencies

This document has two preceding documents, D8.1a, in which the methodology used to define and validate generic user scenarios in vf-OS project is described. D8.1b, which

describes how vf-OS partners have applied this methodology in the definition of the pilots vApps. D8.1c, and D8.1d will describe the development process of the vApps and present the validation process applied to verify if the vApps development respond to user needs and expectations.

0.8 Glossary and Abbreviations

A definition of common terms related to vf-OS, as well as a list of abbreviations, is available in the supplementary and separate document “vf-OS Glossary and Abbreviations”.

Further information can be found at http://www.vf-OS.eu/glossary.

0.9 External Annexes and Supporting Documents

This document has the following external annexes:

- **Annex C:** Pilot 1 Evaluation Report
- **Annex D:** Pilot 2 Evaluation Report
- **Annex E:** Pilot 3 Evaluation Report

0.10 Reading Notes

None

1 Evaluation Roadmap

The evaluation activities conducted in the pilots have two main objectives. The first objective is to verify that the developed software meets the specifications set in the project. In terms of the value proposition for the different stakeholders, this means that Software Developers must be able to develop vApps easily using the vf-OS Open Application Kit (OAK). It also means that developed vApps must respond to real needs in manufacturing environments, as requested by Manufacturers and Logistics Users. The other main objective is to support the Agile methodology adopted in the project for the development of vf-OS Platform components. With this methodology, pilot owners must test the software and provide feedback to developers, so that they can effectively adapt the software to the strict industrial environments regarded. Therefore, to be consistent with this methodology, the evaluation occurs at different stages of the project.

To meet both objectives, vf-OS follows an incremental approach towards evaluation activities. At earlier stages of the project (as this reporting period), many vf-OS Platform Components are not mature enough to make a fair evaluation of the project objectives (specifically, the first release of the design-time components that comprise the OAK are first released in this reporting month). However, some components, ie Middleware, Storage, Data Analytics, Harmonisation, Enablers Framework, or the IO Toolkit are mature enough to test some outstanding vf-OS Platform features and continue their development following an Agile methodology. From this perspective, it is more effective for the project to focus the first evaluation activities on testing critical platform features and collect evaluation data, using the first available component prototypes, rather than trying to validate the usefulness of the vApps or the platform itself.

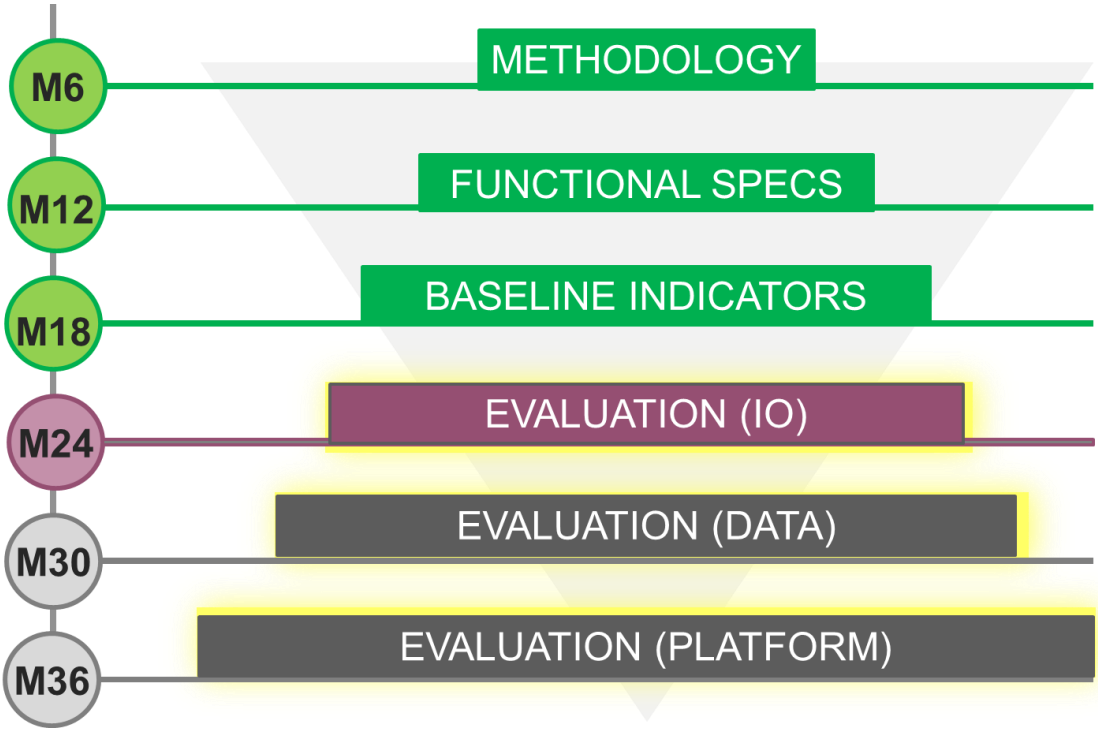


Figure 1: Evaluation roadmap

Figure 1 presents an overview of all the activities conducted in the project piloting work package (WP8). First, the project partners defined the methodology to guide evaluation

activities and agreed on the functional specifications for the pilot applications. From there, the main indicators to validate the applications were defined together with the measurement procedures to obtain them. Then, the validation activities have been divided into three main periods corresponding with the three evaluation reports to be delivered:

- **Evaluation of IO Components and Enablers Framework:** Interaction with manufacturing assets. The first evaluation period is focused on evaluating the capability of the vApps to interact with the manufacturing environment, via components that interact with manufacturing equipment (ie Device Drivers), legacy Software and file formats (ie API Connectors), and specific enablers (vf-OS Enablers) from open source platforms such as FI-WARE
- **Evaluation of Data Management and Processing Components:** Data management and processing. The second evaluation period addresses the evaluation of vApps with an emphasis on their data processing and storage functionalities. The second iteration of Process Designer, Data Mapping, Process Execution, Storage, Transformation and Analytics vf-OS components assure a maturity level where these functionalities can be tested
- **Evaluation of the vf-OS Platform:** vf-OS Platform Evaluation. The final evaluation considers all aspects of the platform, but the emphasis will be on the horizontal components, such as the vf-Store, the vf-Platform, the vf-OAK, and vf-Studio. All of them are wrapping and integrating other vf-OS components and are the main components that vApps developers and users will make use of

In every phase, a subset of the use cases is selected to conduct the evaluation. As explained in D8.1a, this allows developers, at early stages of the project, to focus on some use cases to build early versions (Minimum Viable Products) of the selected vApps from the first platform component prototypes and start collecting data from the pilots. The use cases selected for the first evaluation period use the first available run time components needed to interact with manufacturing assets (delivered in M18). Subsequently, the use cases in the next evaluation periods, use more components as there are more prototypes made available for testing. The next section describes the dependencies between components and use cases in the different evaluation periods. This mapping helps developers to establish priorities for the different integration tasks in the different pilots.

2 Mapping of Development and Evaluation Activities

This section presents the mapping of vf-OS Platform Components and pilot vApps that guides the installation and integration of vf-OS Platform Components in each pilot. D8.1a presented the calendar for the evaluation of the different use cases. The calendar highlighted three main releases at M24, M30, and M36, in accordance with the evaluation roadmap presented in the introduction.

From this calendar, vApp developers and vf-OS Platform Component developers analysed the specifications of every vApp to identify the functionalities needed from the vf-OS Platform in every use case. The result of this analysis is a collection of detailed maps between vf-OS Platform Components and use case releases (one map per pilot) which describes the functional dependencies between pilot use cases and vf-OS Platform Components. Furthermore, these maps, which are detailed in the following subsections, deliver:

- Development roadmaps for pilot vApps. Through these maps, use case development activities are synched, in a time basis, with the development of vf-OS Components. This assures the viability of the development and validation of use cases, by taking into consideration both the functional dependencies between vApps and vf-OS Components and the time constraints between successive vf-OS component releases
- Priorities for vf-OS Component functionalities. Since each pilot vApp should deliver an explicit and legitimate functionality, the development roadmap for vf-OS Components should consider the impact prioritisation with regards to the software specification previously detailed. Through the maps, pilot owners prioritise functionalities, so that vf-OS Component developers can adjust the functionalities of the different releases according to their specific needs

The mapping is presented in a matrix where each row represents a vf-OS Platform Components and each column represents a vApp. The rows include both Design Time and Run Time vf-OS Platform Components (as defined in D2.1 Global Architecture Definition), as well as a list of all Device Drivers, API Connectors and Enablers to be used in the pilots. With this matrix, pilot owners and developers have a shared model that indicates which are the functionalities needed from every component at every phase and this input is used within every technical work-package to guide development and integration activities. In some cases, the available prototypes might not be fully implemented yet and all the functionalities needed to implement the use case under evaluation might not be available, especially during the first evaluation phases. In such cases, vApps and vf-OS Platform Components developers analyse the requirements to agree on a version of the use case that can be already used for evaluation and testing and that will be developed incrementally until the requirements are met.

Finally, a special consideration while reading the mappings in the following sections is that, though vApps were mapped against a complete list of vf-OS components, for readability purposes mapping figures can vary in length. An example is the Pilot 1 mapping (Figure 2) where FIWARE enablers were collapsed in one line as any of them is foreseen to be used in this pilot.

2.1 Pilot 1 Mapping

Figure 2: Pilot 1 Component Mapping

In the case of Pilot 1, as depicted in Figure 2:

- M24 considers vApp12 (vfFailureManager) R1 with a focus on the IO Toolkit, the OPC UA Driver, the Security Components, and the integration with the Messaging component
- M30 considers vApp12 (vfFailureManager) R2 along with vApp11 (vfFailurePrevention) R1, and vApp14 (vfProductionFollowUp) R1 all of them considering first releases of the SDK, vf-Studio, Frontend environment, Process Designer, Process Execution, and Storage
- M36 considers vApp12 (vfFailureManager) R3, vApp11 (vfFailurePrevention) R2, vApp14 (vfProductionFollowUp) R2, together with vApp13 (vfStockPolicies), and vApp15 (vfMaintenanceCalendar). The final development and validation of these vApps will be done using the most up-to-date versions of vf-OS components and vf-OS enablers

2.2 Pilot 2 Mapping

The mapping between CONSULGAL vApps and vf-OS components is shown in Figure 3:

- The M24 release comprehends the delivery of vApp24 (vfConcreteFeedback) R1 and its integration with several vf-OS components namely: Storage, Enablers Framework, and vf-OS Specific Enablers
- With regards to M30 release vApp24 (vfConcreteFeedback) should receive new development updates to increase the functionality delivered (R2) by consuming other vf-OS components namely Platform, vf-Store, vf-Studio, Frontend Environment and Security. Along with this vApp, vApp22 (vfSteelValidation) R1 uses the same vf-OS components as well as an external third-party service provided by the platform and focused on image recognition. Apart from the described interactions, this vApp is used on two more components: Messaging and Storage. In M30, vApp21 (vfDocumentPortal) R1 focus its first release on vf-Studio, Frontend Environment, Messaging, Storage, Security, and the vf-OS Specific Enablers by using the Enablers Framework
- M36 considers the latest release of the vApps along with vApp23 (vfOnSiteManager) R1 and vApp25 (vfProductValidation) R1 where the development and validation of these vApps will be performed by using the most up-to-date versions of each vf-OS component

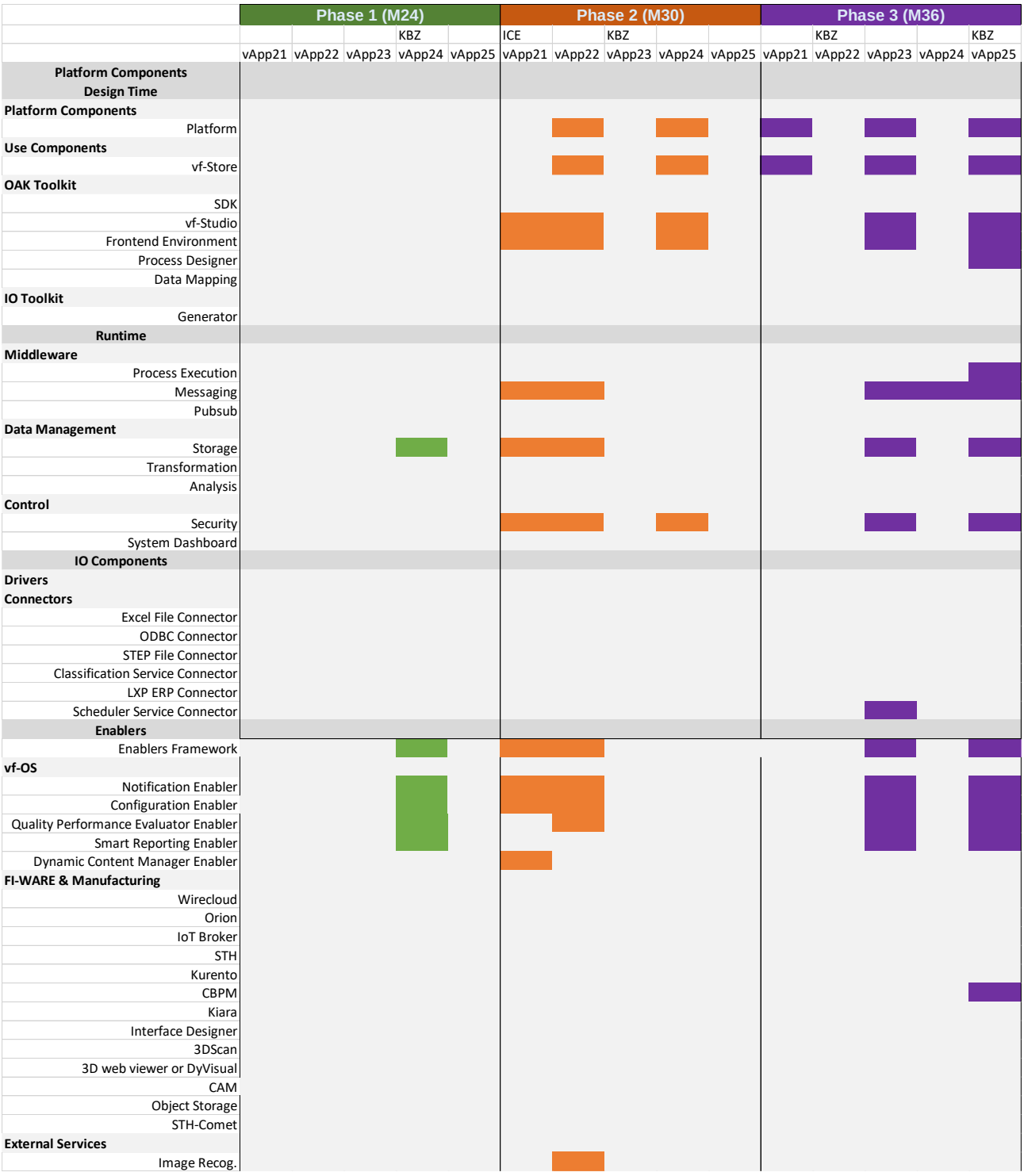


Figure 3: Pilot 2 Component Mapping

2.3 Pilot 3 Mapping

Pilot 3 vApps are released in three phases M24, M30 and M36:

- Phase 1 (M24) concerns vApp31 (vfCollaborationAnalyser) first version orchestrating all the necessary services provided by API connectors, the selected vf-OS specific

enabler and the selected Manufacturing Enablers. The front-end layer is released with the minimal development effort for testing and validation purposes

- Phase 2 (M30) concerns vApp32 (vfIndusEnabler) and vApp33 (vfProductionPlanner) to be released with all necessary vf-OS components and enablers. This phase validates the vf-OS platform integration capabilities and its suitability as a container for business applications. vApp31 (vfCollaborationAnalyser) will be upgraded at the same integration level
- Phase 3 (M36) concerns vApp34 (vfQualityAssurance) and vApp35 (vfProductionTracker) to be released using the most mature vf-OS components and enablers. Additional vApps versions for vApp31, vApp32 and vApp33 will be released during this phase

During these three phases, APR and TARDY will provide their feedback regarding the relevance of released business applications as well as the reliability of proposed technologies. Figure 4 shows the components used in each vApp release.

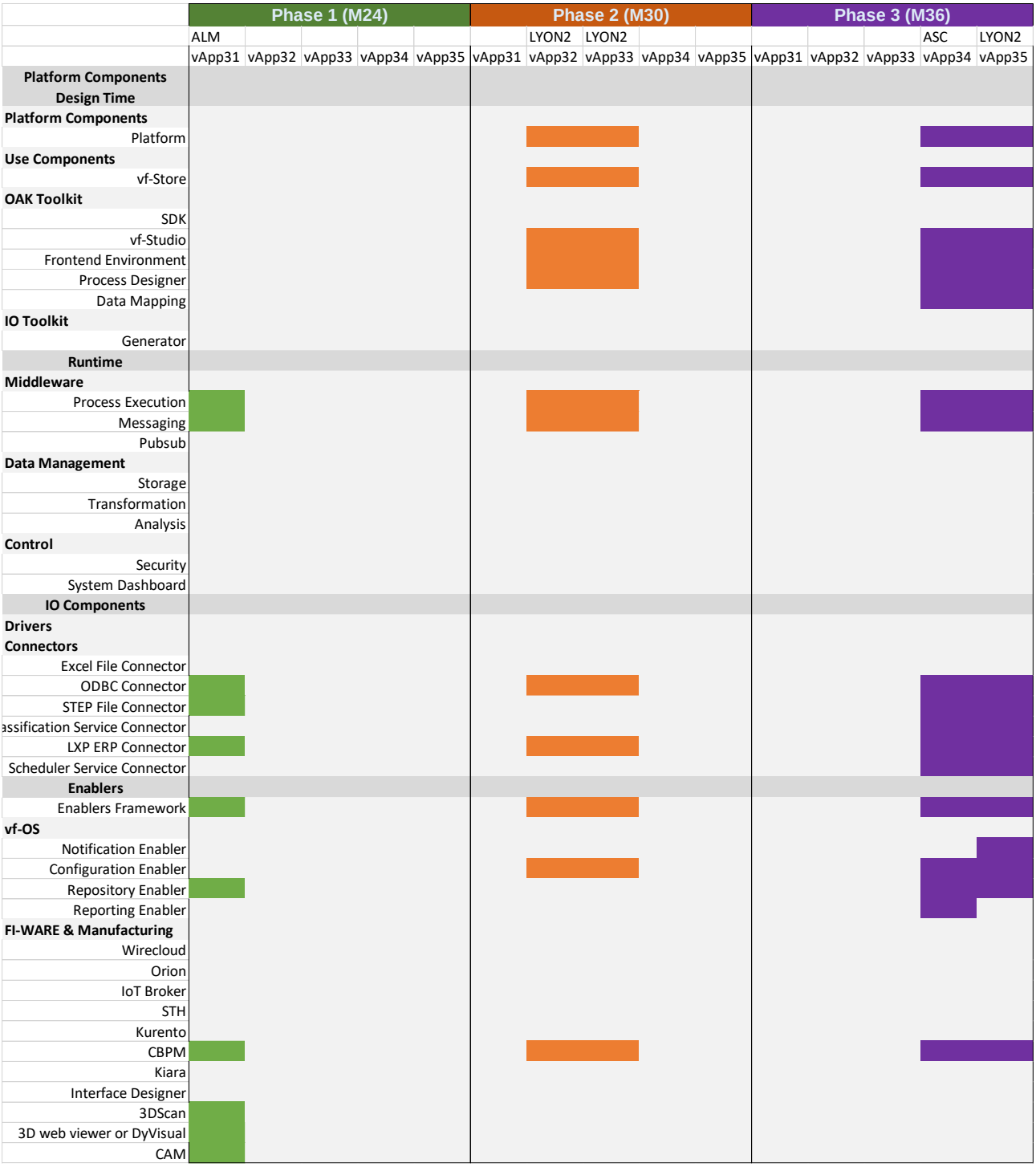


Figure 4: Pilot 3 Component Mapping

2.4 Completeness Check

From the previous mapping, an aggregated coverage of vApps functionality has been made in Figure 5. These components are grouped according to the vf-OS architecture functional blocks introduced in D2.1.

Some of the conclusions that can be extracted from the coverage analysis are:

- All vApps require the use of at least 12-15 vf-OS components and at least one component per functional building block, meaning that vApp developers will require a clear picture of main building components when developing apps for vf-OS
- SDK, Studio and Process Designer are going to be validated significantly, as they are involved in most vApps
- vf-OS specific enablers are used in all the vApp scenarios, and their usage for post-project vApp developers is clearly foreseen
- Functional completeness could be achieved in some specific cases using FI-WARE enablers and external services. The use of external services, although limited in the set vApps, is foreseen to be an empowering tool when developing vApps integrating services from third-party business providers in dynamic and distributed scenarios. WP7 tackles the relevant external services considered by vf-OS as noteworthy

	Pilot 1					Pilot 2					Pilot 3				
	IKERLAN	UPV	CMS	UPV	CMS	ICE	KBZ	KBZ	KBZ	KBZ	ASC	LYON2	LYON2	ALM	LYON2
	vApp11	vApp12	vApp13	vApp14	vApp15	vApp21	vApp22	vApp23	vApp24	vApp25	vApp31	vApp32	vApp33	vApp34	vApp35
Platform Components Design Time															
Platform Components															
Platform	■	■	■	■	■	■	■	■	■	■		■	■	■	■
Use Components															
vf-Store	■	■	■	■	■	■	■	■	■	■		■	■	■	■
OAK Toolkit															
SDK	■	■													
vf-Studio	■	■	■	■	■	■	■	■	■	■		■	■	■	■
Frontend Environment	■	■	■	■	■	■	■	■	■	■		■	■	■	■
Process Designer	■	■	■	■	■					■		■	■	■	■
Data Mapping														■	■
IO Toolkit															
Generator	■	■	■	■	■										
Runtime															
Middleware															
Process Execution	■	■	■	■	■					■	■	■	■	■	■
Messaging	■	■		■		■	■	■	■	■					
Pubsub	■	■	■	■	■										
Data Management															
Storage	■	■	■	■	■	■	■	■	■	■					
Transformation															
Analysis	■	■													
Control															
Security	■	■	■	■	■	■	■	■	■	■					
System Dashboard															
IO Components															
Drivers															
TS OPC UA Driver	■	■	■	■	■										
Other Drivers															
Connectors															
Excel File Connector			■	■	■										
ODBC Connector											■	■	■	■	■
STEP File Connector											■			■	■
Classification Service Connector														■	■
LXP ERP Connector											■	■	■	■	■
Scheduler Service Connector								■	■					■	■
Enablers															
Enablers Framework	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
vf-OS															
Notification Enabler		■	■	■	■	■	■	■	■	■					■
Configuration Enabler	■		■	■	■	■	■	■	■	■		■	■	■	■
Quality Performance							■	■	■	■	■			■	■
Evaluator Enabler							■	■	■	■	■			■	■
Smart Reporting Enabler								■	■	■				■	
Dynamic Content Manager Enabler						■									
FI-WARE & Manufacturing															
Kurento											■	■	■	■	■
CBPM										■					
Kiara															
Interface Designer											■				
3DScan											■				
3D web viewer or DyVisual											■				
External Services															
Image Recog.							■								

Figure 5: Pilots coverage

3 Evaluation Roles and Guidelines

3.1 Evaluation Roles

The methodology in D8.1a introduced roles involved in the evaluation and validation of vApps:

- **Manufacturing and Logistic Users:** The customers and consumers of vApps
- **Software Developers:** vApps developers with the necessary capabilities to implement Manufacturing and Logistic Users' needs
- **Evaluators:** The neutral party responsible for evaluation based on data gathered from Manufacturing and Logistic Users and Software Developers

According to these definitions, the table of Figure 6 shows the partners involved in each pilot and the roles that each will take in the evaluation of every pilot:

- **Pilot 1:** UPV, IKERLAN, and CMS are Software Developers of the manufacturing vApps. MASS is playing the role of Manufacturing User, and UNINOVA, UPV, IKERLAN, and CMS are the Evaluators in charge of gathering data from the vApps evaluations. UNINOVA provides the evaluation guidelines for the rest of the partners and conducts the evaluation in the first evaluation period of each pilot. For the rest of the use cases, UPV, CMS, and IKERLAN exchange the roles of Software Developer and Evaluator to balance the workload
- **Pilot 2:** ICE and KBZ are Software Developers of the manufacturing vApps. CONSUGAL is playing the Manufacturing User, and UNINOVA takes the Evaluator role for all the vApps evaluations
- **Pilot 3:** LYON2, ASC, and ALM are Software Developers of the manufacturing vApps. APR and TARDY are playing the manufacturing user roles, and UNINOVA takes the evaluator role for all the vApps evaluations

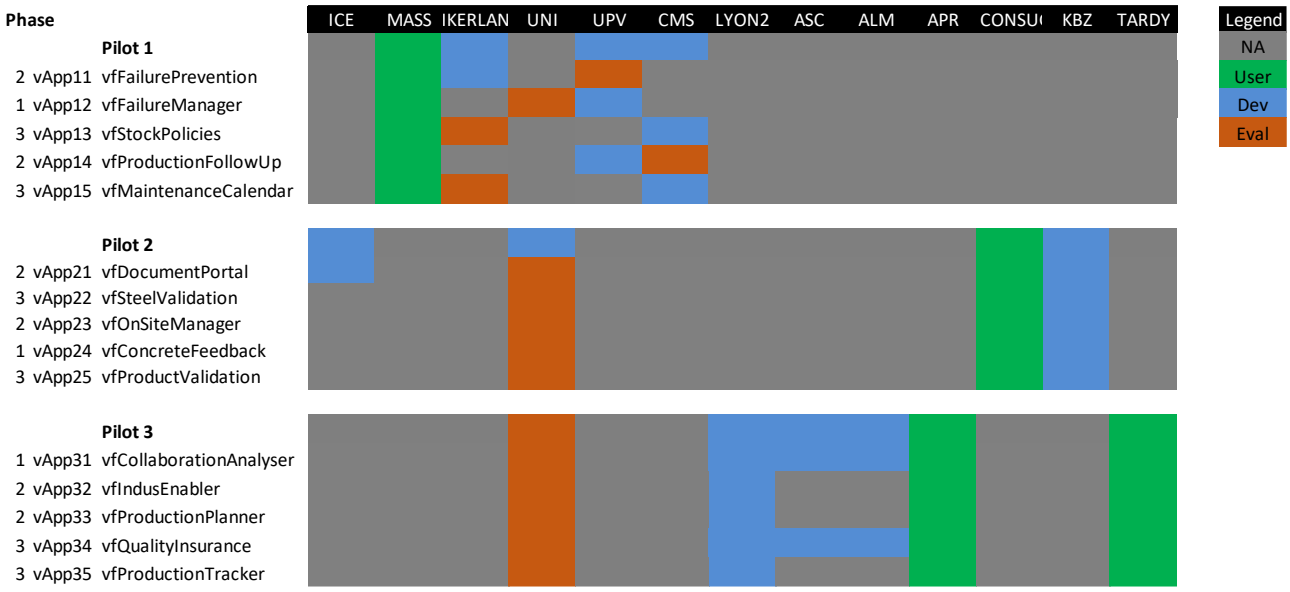


Figure 6: Pilots evaluation roles

3.2 Evaluation Guidelines

This section provides guidelines to conduct the evaluation of all pilots. It follows the methodology illustrated at Figure 7 that is composed of two main phases: the evaluation questionnaires preparation and the evaluation execution. The first main phase comprises the definition of two questionnaires, specifically related to generic/overall and specific pilot information. To provide a more complete understanding and feedback of the object being evaluated, both questionnaires comprise a combination of qualitative and quantitative questions.

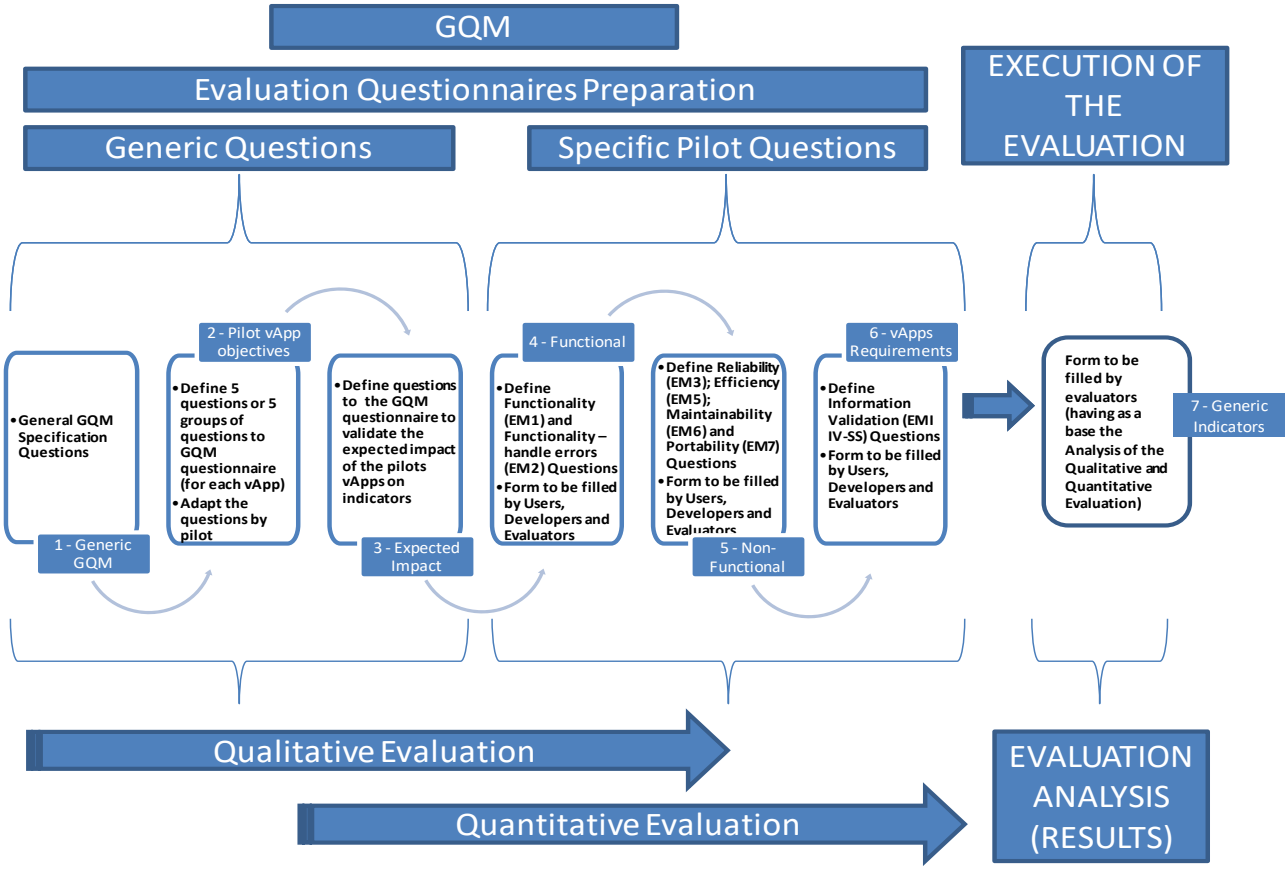


Figure 7: GQM-based Evaluation Methodology

The Generic Questions are designed to reinforce the understanding of the applications (vApps) needs and requirements to Manufacturing and Logistic Users and Software Developers. These questions are also designed to verify if each specific pilot questionnaire is using the identified metrics, and consequently evaluating its specific goals and each vApp requirements. Thus, since this evaluation follows the GQM method (Goal, Question, Metrics, BCR04), the questions identify key points that are critical for the assessment of the defined goals. Then, the metrics provide valuable context for the values to be used in the evaluation questions. The defined Generic Questions includes characteristics such as functionalities, advantages, cost, popularity, technology, level of the reliability/portability/satisfaction of the vApp, etc., related to the vApps to be evaluated (the questionnaires are available in the annexes C; D and E of D8.1c).

The Specific Pilot Questions intend to conduct vApps evaluations at the pilots, which follows a standard-based approach. Due to its complexity and limited resources in this task the followed methodology does not follow the mentioned standards. However, it tries to

evaluate the vApps in all the standards addressed characteristics. This second standard-based phase of the Software Evaluation Methodology was defined having in consideration the following standards.

- **ISO/IEC CD 25010** for the quality model determining which quality characteristics will be considered when evaluating the properties of a software product This is in relation to whether the system satisfies the requirements of its several stakeholders, and thus provides value. The product quality model defined in ISO/IEC CD 25010 comprises the following quality characteristics: Functional Suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability and Portability. In this case it is being used six of the eight characteristics of this standard to measure the product quality. The exceptions are the compatibility and security characteristics. However, the security is a fundamental feature considered in the project having a technical team ensuring its implementation and validation. In relation to the compatibility, the entire project has its platform and solutions designed to assure such characteristics namely in relation to interoperability and co-existence features, which are its sub-topics
- **ISO/IEC 25030** is part of the Software product Quality Requirements and Evaluation (SQuaRE) series of International Standards (25030:2007). This standard provides requirements and recommendations for the specification of software quality requirements. Software product quality requirements are needed for specification, planning, development, and evaluation. This standard provides requirements and recommendations for product quality assurance. It was followed in the development pipeline set in the project. Thus, it is related to the evaluation after the software development of the vApps, which is preceded by specification and planning activities
- **ISO 9241** is a multi-part standard from the International Organization for Standardization (ISO) covering ergonomics of human-computer interaction. In this case, Parts 12-17 of this standard which deal with usability aspects of user interfaces were used

The GQM-based Evaluation Methodology that has been followed integrates seven steps. At this deliverable release, only the three first ones have been conducted (see D8.1c Annexes). Thus, the first three types of generic questions related to each pilot vApp have been already answered. These include questions about generic vf-OS Platform functions that relate each developed vApp to its specific objectives and expected impact.

The following three steps relate to the definition of the Specific Pilot Questions, which compose a set of 8 Evaluation Modules (EM) specifically related to functional, non-functional, and vApps requirements. Figure 8 contains a table with the definition, the description, and the expression of the base indicators used in each EM (this information was originally provided in D8.1a and is included here for the sake of clarity). There will be an evaluation form for each vApp designed to collect the required information to calculate these metrics. Figure 8 also indicates which stakeholder must answer the questions in the evaluation form for each EM (ie who is the EM Evaluation Form stakeholder).

EM1 Functionality Metrics		
Name	Description	Expression
Functionality Metric of each Scenario Step **(FMSS)**	Number of successful step executions (Suc_{cess}) divided by the Total number of Executions (TSS_n)	$FMSS = \frac{TSS_{success}}{TSS_n}$
EM1 Functionality Metric of the Pilot **(FMT1)**	Average of the FMSS over the total number of scenario steps (SS_n)	$FMT1 = \frac{\sum_{i=1}^{n} FMSS_i}{SS_n}$
EM1 Evaluation Form Stakeholder: **Manufacturing and Logistics Users**		
EM2 Functionality (handle errors)		
Name	Description	Expression
EM2 Functionality Metric of each Scenario Step (handle errors) **(FMSS2)**	Number of successful executions of unit tests handling specific, pre-established errors ($UTSS_{success}$) divided by the total number of unit test executions ($UTSS_n$)	$FMSS = \frac{UTSS_{success}}{UTSS_n}$
EM2 Functionality Metric of the Pilot **(FMT2)**	Average of FMSS2 over the total number of scenario steps (SS_n)	$FMT2 = \frac{\sum_{i=1}^{n} FMSS2_i}{SS_n}$
EM2 Evaluation Form Stakeholder: **Manufacturing and Logistics Users**		
EM3 Reliability		
Name	Description	Expression
EM3 Reliability for each Scenario Step **(RMSS)**	Number of faults (F) detected when performing pilot scenario step executions divided by the total number of executions (TSS_n)	$RMSS = \frac{F}{TSS_n}$
EM3 Reliability Metric of the Entire pilot **(RMT)**	Average of RMSS over the total number of scenario steps (SS_n)	$RMT = \frac{\sum_{i=1}^{n} RMSS_i}{SS_n}$
EM3 Evaluation Form Stakeholder: **Software Developers**		
EM4 Usability		
Name	Description	Expression

EM4 Usability Metric for each Scenario Step **(UMSS)**	The user interface of the software is evaluated in accordance with the software related parts 12 to 17 of IS0 9241 [ISO+01], which are respectively related to:1) Presentation of information; 2) User guidance; 3) Menu dialogues; 4) Command dialogues; 5) Direct manipulation dialogs; 6) Form filling dialogues. The usability metric (UMSS) for each scenario step will be the number of successful Answers ($A_{success}$) divided by the number of Check Lists (CL_n)	$UMSS = \frac{A_{success}}{CL_n}$
EM4 Usability Metric of the Pilot **(UMT)**	Average of UMSS over the total number of scenario steps (SS_n)	$UMT = \frac{\sum_{i=1}^{n} UMSS_i}{SS_n}$
EM4 Evaluation Form Stakeholder: **Manufacturing and Logistics Users**		
EM5 Efficiency		
Name	Description	Expression
EM5 Efficiency Metric for each Scenario Step **(EMSS)**	Sum of Efficiency Answers scores (AU) to the Efficiency questions given to classify the efficiency of the scenario step (U_n is the total number of questions)	$EMSS = \frac{\sum_{i=1}^{n} AU_i}{U_n}$
EM5 Efficiency Metric of the Pilot **(EMT)**	EMSS of each scenario step divided by the total of scenario steps (SS_n)	$EMT = \frac{\sum_{i=1}^{n} EMSS_i}{SS_n}$
EM5 Evaluation Form Stakeholder: **Manufacturing and Logistics Users**		
EM6 Maintainability		
Name	Description	Expression
EM6 Maintainability Metric of each Scenario Step **(MMSS)**	Sum of Maintainability Answers scores (MA) to the Maintainability questions given to classify the efficiency of the scenario step (MQ_n is the total number of questions)	$MMSS = \frac{\sum_{i=1}^{n} MA_i}{MQ_n}$
EM6 Maintainability Metric of the Pilot **(MMT)**	Average of MMSS over the total number of scenario steps (SS_n)	$MMT = \frac{\sum_{i=1}^{n} MMSS_i}{SS_n}$
EM6 Evaluation Form Stakeholder: **Software Developers**		
EM7 Portability		
Name	Description	Expression
EM7 Portability Metric of each Scenario Step **(PMSS)**	Average of the component portability metric (PMC) over the total number of step components (C_n)	$PMSS = \frac{\sum_{i=1}^{n} PMC_i}{C_n}$
EM7 Portability Metric of the Pilot **(PMT)**	Average of PMSS over the total number of scenario steps (SS_n)	$PMT = \frac{\sum_{i=1}^{n} PMSS_i}{SS_n}$

EM7 Evaluation Form Stakeholder: **Software Developers**		
EMIV Scenario Step Information Validation		
Name	Description	Expression
EMIV Efficiency Metric of the Pilot **(EMIV)**	This evaluation process considers the different vApps in every pilot. The measurement of the information validation of each pilot (EMP) will be the number of successful addressed characteristics (project scenario requirements) divided by the total number of such characteristics (CH_n)	$EMIV = \frac{\sum_{i=1}^{n} EMP_i}{CH_n}$
EMIV Functionality Metric of the Pilot **(EMIVP)**	Average of EMIV over the total number of scenario steps (SS_n)	$EMIVP = \frac{\sum_{i=1}^{n} EMIV_i}{SS_n}$
EMIV Evaluation Form Stakeholder: **Evaluators**		

Figure 8: Pilot EM metrics and stakeholders

The last step of the GQM-based Evaluation Methodology relates to the analysis of the evaluation conducted. The evaluation results can be used to improve the pilots vApps, sustain positive outcomes, and provide support for errors prevention. Thus, with the writing of the evaluation reports and disseminating them to both Manufacturing and Logistic Users and Software Developers it is possible to determine meaningful programmatic changes based on evaluation findings, if necessary.

The act of evaluating the vApps will happen in three different iterations (M26; M30 and M36) accordingly to the vApps development scheduling illustrated in the Figure 9. Such evaluations will happen at a common place during the project meetings at the final phase of each iteration with all the evaluation stakeholders together following the forms for each vApp. The analysis of the results will be reported in the following days by the Evaluators.

Thus, at the moment of this deliverable it was accomplished all the three first steps (Generic Questions definition) for all the vApps. Then, all the subsequent steps (from 4th to 7th) for each vApp developed will be conducted in each development iteration.

Figure 9: vApps Development Scheduling

4 Conclusions

Deliverable “D8.1c Validation Scenarios” and its annexed documents describe in detail the evaluation and validation activities that will be conducted using the pilot applications (vApps) defined in the different validation scenarios.

The document presents the conclusions of intensive meetings where vApp developers analysed along with pilot manufacturing users the deliverable D8.1b. In summary, the mapping between pilot vApps and vf-OS components is articulated with the perspective to integrate gradually available vf-OS components in parallel with the development of specific pilot vApps. This requires a high level of communication between pilot owners, vApp developers, and vf-OS Components developers, to explain and fine-tune every use case feature. From these interactions, vApp developers selected the vf-OS components that they plan to use when developing their respective vApps. As far as vf-OS components are iteratively developed, vApp developers faced the challenge to provide a feasible calendar of vApp releases. The mapping and alignment between vApps and vf-OS components has been introduced in Sections 2.1, 2.2, and 2.3. A conclusive analysis of the completeness coverage of user needs is done in Section 2.4.

Mappings establish agreements between vApp developers and vf-OS Platform Component developers aligning respective releases an assuring the delivery of vApp versions to pilot users. The result from a user perspective is a completeness coverage of their needs through the proper integration and usage of vf-OS components.

Additionally, the document includes the role assignment to vf-OS partners under the evaluation methodology GQM provided in deliverable D8.1a. Section 3.2 where the next steps of the methodology in terms of vApp evaluation is outlined.

The evaluation to be conducted in three periods (M24, M30 and M36) is documented per pilot basis in the following annexes:

- **Annex C: Pilot 1 Manufacturing and Logistics / Automation.** Applications focused on providing support to maintenance and spare part stock management in automation production equipment
- **Annex D: Pilot 2 Construction / Industrialisation.** Applications to improve time and resource utilisation in construction projects through advanced collaboration between construction managers and material providers
- **Annex F: Pilot 3 Manufacturing Assembly / Collaboration.** Applications to enable collaborative design and manufacturing of complex products

Annex A: History

Document History	
Versions	V0.0.1: • Table of context and first draft version V0.1.1: • Sections 2.1, 2.2 and 2.3 pilot mappings V0.1.2: • Sections 2.4 added V0.1.3: • Sections 2.4 completed. • Improvements to section 2.1, 2.2 and 2.3 V0.1.4: • Several fixes on conclusions and section 3.1 V0.1.5: • Finished section 2.2, changes in section 3.1 V0.1.6: • Finished section 3.2 V0.1.7, v0.1.8: • First internal review V1.0.1: • Second internal review V1.0.2, v1.0.3: • Coordinator review
Contributions	UPV: • Francisco Fraile, Victor Anaya UNINOVA: • Joao Sarraipa, Elsa Jesus KBZ: • David Aleixo, Carlos Agostinho (1st internal review) CMS: • Carlos Coutinho LYON: • Nejib Moalla IKERLAN: • Eduardo Saiz (2nd internal review) ICE: • Óscar García (coordinator review)

Annex B: References

Basili, V., Caldiera, G., & Rombach, D. (1994). In: “The goal question metric approach” Export Find Similar, edited by: Marciniak, J., published by Encyclopaedia of Software Engineering (1994).

Annex C, D, E: Evaluation Reports

The Evaluation Reports are handed as separate files. The series of annexed documents are named:

- D8.1c: Annex C: Pilot 1 Evaluation Report
- D8.1c: Annex D: Pilot 2 Evaluation Report
- D8.1c: Annex E: Pilot 3 Evaluation Report

These annexed documents can be found in the same folder as this document and with the appropriate version. This separation is made to help the reader managing the Use Case descriptions since they are more clearly presented grouped per project pilot.

www.vf-OS.eu