ZDMP: Zero Defects Manufacturing Platform

WP4: Technical Challenge: Requirements, Specifications, and Standardization

D4.1a: Requirements Document and Update - Vs: 1.0.1

Deliverable Lead and Editor: Radu Lazar, SIVECO

Contributing Partners: All Partners

Date: 2019-09

Dissemination: Public

Status: Approved

Abstract

The requirements of the ZDMP project, this document, provide an essential tool for fixing the project scope by eliciting the needs of pilots and software developers as final users of ZDMP applications, zApps. Expectations and assumptions are included to reflect what the project will do and not do. It is also a critical source of information for the design, development and implementation activities to be carried out as well as to validate the project outcomes.

Grant Agreement: 825631

Document Status

Deliverable Lead	Radu Lazar, SIVECO
Internal Reviewer 1	Ronal Bejarano Rodriguez, TAU
Internal Reviewer 2	Marc Dorchain, SAG
Internal Reviewer 3	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP4: Technical Challenge: Requirements, Specifications, and Standardization
ID	D4.1a: Requirements Document and Update
Due Date	2019-06
Delivery Date	2019-09 (v1.0.0) & 2020-02 (v1.0.1) Due to Review request
Status	Approved

Project Partners:

For full details of partners go to www.zdmp.eu/partners

Executive Summary

The main purpose of this deliverable “ZDMP D4.1a - Requirements Document and Update” is to collect the requirements to be provided by the ZDMP ecosystem (ZDMP platform, SDK, and applications “zApps”). This deliverable is the result of the activities performed in task T4.1 (Requirements Document and Updateof the project which succeeded to achieve the objective O4.1 to define ZDMP requirements, specified in the Description of Action (DOA) for the development of WP4 - “Technical Challenge: Requirements, Specifications, and Standardization”. Implementing these requirements will guarantee the achievement of ZDMP’s scope.

This task primarily includes the analysis of the document D2.3 - Industry Scenarios and Use Cases (T2.3) resulting in a list of user/system needs and features for the industrial pilot implementations. This information will be compared to the ZDMP vision, defined in T2.1 (Vision Consensus), analysed, mapped, and aligned to other Zero-defects Manufacturing Reference Models (T2.4). Based on the aforementioned, functional, and non-functional requirements were generated for the ZDMP platform, ZDMP SDK, and applications (“zApps”) proposed in the D2.3 document.

To achieve the objectives of T4.1 the following actions were conducted:

- **Define** the methodology used to formulate the functional and non-functional requirements for the development of the ZDMP Platform
- **Extract** the needs and requirements provided by all the thirteen industrial pilots participating in the project
- **Produce** a functional requirements list that defines the user needs and features of the ZDMP components
- **Produce** a non-functional requirements list (performance, operational, maintainability, security, log, and audit, “look and feel”, constraint, availability, etc.) necessary to be considered were formulated

These actions together, resulted in a complete list of requirements (See Annex C) grouped into the following categories:

- **User requirements**: Provided by all the thirteen pilots (zApps users) and by the zApps software developers. This enable the verification of the user needs by the development team. The requirements found are derived from the project vision and can be traced back to the project goals and objectives
- **System requirements**: Identification and description which facilitate the definition of the architectural structure for satisfying the “User” requirements
- **Software requirements**: Developer topics needed to implement the “User” requirements either those for the pilots or those for the intended commercial exploitation actions

This deliverable is the basis for the development tasks to be conducted in WP4 to WP10.

This v1.0.1 version document includes the implementation of changes following the projects M9 Review.

Table of Contents

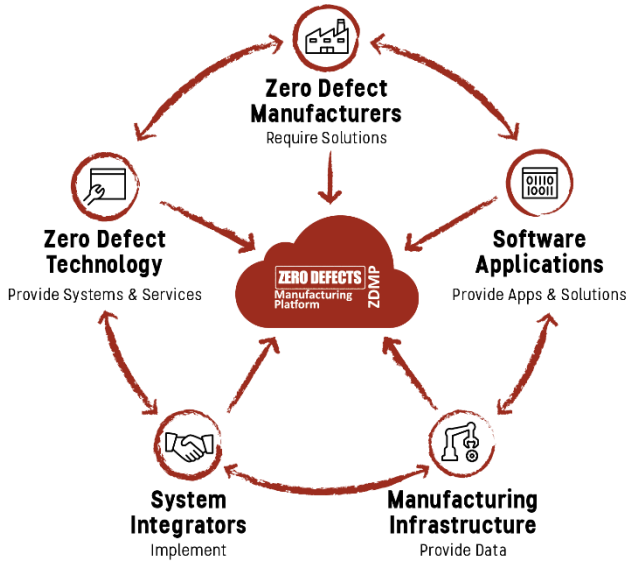
0 Introduction....1
1 Scope and Methodology....5
1.1 Introduction....5
1.2 Objectives....6
1.3 Methodology to Produce the Deliverable....7
1.3.1 Inputs Analysis....7
1.3.2 Definition of Requirements Sources....8
1.3.3 Definition of Requirement Levels and Traceability....9
1.3.4 Best practices for Requirements Post-processing....10
1.4 Literature analysis....11
1.5 Main Results and Outputs....11
2 Elicitation Activities....12
2.1 Strategy....12
2.1.1 User Requirements....12
2.1.2 System requirements....13
2.1.3 Software requirements....13
2.2 Guidelines to Perform the Requirements Elicitation....13
2.2.1 Analysis of the Scope of Requirement Types....13
2.2.2 Requirements Elicitation Best-Practices....14
2.3 Requirements Scope....15
2.4 Analysis and Post-Processing....16
3 Requirements Specification....17
4 Conclusions....19
Annex A: History....20

0 Introduction

0.1 ZDMP Project Overview

ZDMP – Zero Defects Manufacturing Platform – is a project funded by the H2020 Framework Programme of the European Commission under Grant Agreement 825631 and conducted from January 2019 until December 2022. It engages 30 partners (Users, Technology Providers, Consultants and Research Institutes) from 11 countries with a total budget of circa €16.2 million. Further information can be found at www.zdmp.eu.

In the last five years, many industrial production entities in Europe have started strategic work towards a digital transformation into the fourth-industrial revolution termed Industry 4.0. Based on this new paradigm, companies must embrace a new technological infrastructure, which should be easy to implement for their business and easy to implement with other businesses across all their machines, equipment, and systems. The concept of zero-defects in the management of quality is one of the main benefits deriving from the implementation of Industry 4.0, both in the digitalisation of production processes and digitalisation of the product quality.



To remain competitive and keep its leading manufacturing position, European industry is required to produce high-quality products at a low cost, in the most efficient way. Today, the manufacturing industry is undergoing a substantial transformation due to the proliferation of new digital and ICT solutions, which are applied along the production process chain and are helping to make production more efficient, as in the case of smart factories. The goal of the ZDMP Project is to develop and establish a digital platform for connected smart factories, achieving excellence in manufacturing through zero-defect processes and zero-defect products.

ZDMP aims at providing such an extendable platform for supporting factories with a high interoperability level, to cope with the concept of connected factories to reach the goal of zero-defect production. In this context, ZDMP will allow end-users to connect their systems (ie shop-floor and Enterprise Resource Planning systems) to benefit from the features of the platform. These benefits include product and production quality assurance, amongst others. For this, the platform provides the tools to allow following each step of production, using data acquisition to automatically determine the functioning of each step regarding the quality of the process and product. With this, it is possible to follow production order status and optimise the overall processes regarding time constraints and product quality, achieving the zero defects.

0.2 Deliverable Purpose and Scope

The main purpose of this document “D4.1a: Requirements Document and Update” is to collect, analyse, and define the requirements for the ZDMP project, ie the ZDMP platform, the ZDMP SDK, and applications(zApps) which define the specific components needed to develop on this project. The elicitation and analysis of needs and requirements have multiple sources:

- The project pilots (D2.3)
- The ZDMP project vision and mission (D2.1))

This deliverable will be updated in a further version of it at the end of the project, to align its content to the business model and the development of ZDMP Limited activities.

Specifically, the DOA states the following regarding this deliverable:

T4.1	**Requirements Analysis**			**SCALE**	**M4-6, 46-48**
D4.1ab	Requirements Document and Update	R	PU	6, 48	RDI1 & 8
The requirements analysis is the borderline between the business/requirements aspects and the software engineering process of the project. The specification will consider the DOA, the functional requirements from the target user and provider scenarios, requirements arising from the preliminary scenarios defined within the pilots, the state-of-the-art analysis, and the partners’ further knowledge and expertise. The deliverable will be a joint document for the requirements of all project components, divided into strategic as well as high level functional and technical requirements. The requirements will focus on the user input and will be strongly driven by users by allowing them to define “User Stories”. This approach is derived from the SCRUM approach, which is an agile development approach, which allows user-centric definition of user stories within a so called “product backlog”. Each user story will define a very small part of the requirements definition and may be as small as one sentence saying “As a [role], I would like to [...]”, eg “As a shop floor manager, I would like to be alerted by ZDMP in case of anomalies in the sensor measurements”. D4.1b also contains an integration and generalisation of the analysis, and describes the generic requirements aimed at sustainability beyond the project’s lifetime based on lessons learnt as the pilots are developed and will come at month 48. They also form an embryo of a product management plan for ZDMP Limited. The deliverable will be a joint document for the requirements of all project components, divided into strategic, high level functional, non-functional, and technical requirements.					

The main results reported in this deliverable can be grouped into three categories:

- User-level needs (or User requirements)
- System Requirements
- Software Requirements

0.3 Target Audience

The document “D4.1a: Requirements Document and Update” is aimed primarily at project developers and describes the user needs that will guide their application development.

0.4 Deliverable Context

This document is a deliverable of Task 4.1 in WP4. WP4 defines the ZDMP technical approach including the overall architecture and the specifications which are needed in work packages WP5-8.

This deliverable has links to other project documents as follows:

Primary Preceding documents:

- **Industry Scenarios and Use Cases (D2.3)**: This document is a reference document focused on defining the top-level requirements associated to the industrial pilots that will be the validation scenarios of the ZDMP project

Primary Dependant documents:

- **User Mock-ups (D4.2):** This document presents graphical designs and early mock-ups of all ZDMP components and platform from a user-perspective
- **Global Architecture Specification (D4.3)**: This document acts as an architectural reference for the technical work packages (WP 5, 6, 7, and 8)
- **Functional Specification (D4.4)**: This document acts as a functional reference for both the technical work packages (WP 5, 6, 7, and 8) as well as the zApps built in the pilots of WP9/10

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1: Scope and Methodology:** Performs a literature analysis and describes the adopted methodology for conducting the whole process of requirements gathering
- **Section 2: Elicitation Activities:** Describes the set of activities that were performed for the execution of the elicitation and collection of requirements
- **Section 3: Requirements Specification:** Describes the actual list of requirements, its structure, and meaning
- **Section 4: Conclusions:** Provides the conclusions of this task's work

- **Annexes:**
  - **Annex A:** Document History
  - **Annex B:** Reference
  - **Annex C:** ZDMP requirements

0.6 Document Status

This document is listed in the Description of Action (DOA) as public access.

0.7 Document Dependencies

The document "D4.1a: Requirements Document and Update", despite being targeted for delivery during the first period of the project, will be a living document, subject to documented changes during the project life cycle since, obviously, the requirements will grow and evolve over time or even some may be removed. These will be tracked internally which will mainly affect the list of requirements defined in Annex C. These will be marked as different versions in the corresponding field, so the users will not have to revise the entire document but only search for the changes concerning each version.

D4.1b is submitted at the end of the project in M48 will include the latest composition and status of the requirements and is aimed at the sustainability beyond the project's lifetime based on lessons learnt as the pilots are developed.

0.8 Glossary and Abbreviations

A definition of common terms related to ZDMP, as well as a list of abbreviations, is available at http://www.zdmp.eu/glossary.

0.9 External Annexes and Supporting Documents

- Annex C: Snapshot of Requirements - Spreadsheet
- Online living Repository: http://www.zdmp.eu/requirements

0.10 Reading Notes

- None

0.11 Document Updates

Following the M9 Review comments the original document (v1.0.0) was requested to be resubmitted (v1.0.1) and the issues raised were addressed as follows:

Issue	How/Where addressed
6) The list of requirements needs to be fixed in a definitive state which is agreed upon by all partners at a point in time. This version has to be resubmitted in SYGMA as a kind of baseline. This does not exclude requirements from changing, but these changes need to be documented. While the requirements might change, the categorization shall remain the same from this fixed state on.	The latest and slightly revised requirements list is resubmitted as the original document plus the XLS attachment. Note that D4.1b is for submission at M48 and was exactly planned for that with changes being documented throughout the period and available within it. We do not fully agree with the point that "categorisation shall remain the same" which presumably means the priority. We of course expect it to be largely the same but this is an agile process as specified in the DOA and some requirements will go up, down, or disappear which is the natural part of a normal agile process in that the changes can be even quite significant. Indeed, this is conformant with your point on changing/documenting requirements, so we assume this is just a small glitch in the text
The document provides the methodology for defining the requirements, but states that it is a living document. The format and contents of the Excel file, used to store the requirements and currently submitted in Sygma is not the one available in the project website.	There were not the same because the excel was still living. However, a post review update has been provided including the current XLS in the package and following other review comments so items have also been modified on priority etc

In addition, the following comments were made which will be addressed in the M18 version of this deliverable

Issue	Comment
None	N/A

1 Scope and Methodology

1.1 Introduction

ZDMP requirements gathering aims to collect all needed information from the customer in this case the pilot users and the potential of Zero Defects limited order to well understand its requirements.

This is why requirements' gathering is one of the main activities to be conducted for the definition of a project's scope and the basis of project design. Its purpose is to elicit, analyse, and establish the user needs. The identified user needs and the analysis of the present situation allow their transformation into the ZDMP framework requirements. Several iterations are performed in order to describe functionalities of the system under development guiding the implementation to fulfil and accomplish the scope of the project.

The requirements address the needs of relevant stakeholders, including the needs that are pertinent to the various product lifecycle phases [CKS11].

The Capability Maturity Model Integration (CMMI) defines the following major steps for Requirements' Development:

- Elicitation of needs, using document analysis and reviews, interviews, questionnaires, and analysis of use-cases and user stories
- Transformation of stakeholder needs into User Requirements
- Refinement of the User Requirements into System and Software Requirements
- Analysis of the requirements, ensuring they are necessary and sufficient
- Balance stakeholder needs and constraints
- Validation of the requirements by the major stakeholders

The methodology used for defining the ZDMP requirements includes all above actions and best-practices, including the analysis of documentation (eg D2.3 - Industry Scenarios and Use Cases), face-to-face meetings, and remote talks between the eliciting team and pilots' stakeholders. It also included strict actions of both converting the acquired knowledge to User Requirements. System requirements are produced as a result of User Requirements analysis. User Requirements in conjunction with System requirements will finally lead to production of Software Requirements. Each of these actions is finalised by a validation from the stakeholders of each industrial pilot, which included checks for completion and sufficiency and validation of the correctness of the requirements.

All observations performed by the stakeholders at this step are analysed, agreed, and implemented in the requirement document

Additionally, the Project Management Institute's Project Management Body of Knowledge (PMI PMBoK) [PMI13] states that requirements should always include a traceability matrix that links the product requirements to their source (and, additionally, to the component implementation that satisfies them).

The layout and the content of Annex C allows the traceability of ZDMP project requirements, including the connection between the User Requirements and its source (eg interview, document), as well as the connection between the User, System, and Software Requirements. This allows a check that all user requirements are covered by software requirements. In future ZDMP tasks, namely in WP4 to WP10, it is expected that this traceability will be extended to trace and interconnect the requirements with product design and development evidences, and with test scenarios and test cases.

The result is an Online living Repository (http://www.zdmp.eu/requirements) for which Annex C also points to an external Excel file which is a snapshot of this provided for the M9 review. An updated deliverable at M48, D4.1b, will both update this document if necessary but primarily include a snapshot of the updated repository at the end of the project where any changes and status assessments will also be given. This will also serve as input to joint (eg ZDL) and individual exploitation plans to help shape further partner activity.

1.2 Objectives

The task T4.1 has to define very precise statements for the scope of the ZDMP project. This will be performed through elicitation, analysis, validation, and registration of customer needs, expectations, constraints, expected market to obtain prioritised user requirements that constitute an understanding of what will satisfy the stakeholders of the ZDMP project (aka “customers”). These customers will be the users of the ZDMP pilots and all potential users from future businesses that may use the ZDMP framework (eg zApps software developers).

The requirements are the basis for the future project’s business objectives, design, and development. They are also involved in specification for change management to deliver development, testing and production environment.

The work performed for ZDMP task T4.1 started by analysing the specific characteristics and needs of the industrial pilots participating in the project, and the relevant feedback information collected in deliverables D2.3 (Industry Scenarios and Use Cases) and D9.1&D10.1 (Implementation Definition, Planning, KPI Validation/Measurement Criteria).

This provided the main industrial needs to be addressed and helped define the set of functional and non-functional requirements to be provided by the ZDMP platform and applications. This deliverable thus reflects the approach taken to define the functionalities to be provided by ZDMP as well as the non-functional requirements to be fulfilled by its different components. The deliverable includes:

- The definition of the methodology used to define the requirements to consider during the development of the ZDMP framework
- The description of the needs and requirements provided by the different industrial pilots participating in the project (ZDMP users) and zApps developers
- The compiled functional and non-functional requirements addressed to the development of ZDMP components

The requirements will not be limited to the users of the ZDMP pilots needs since the final product is open to a large area of industrial clients as well as the sub call experimenters. This is why all requirements will be analysed against to the conclusions of D3.1 “Market Analysis and Competition Report”. This is necessary to enforce that the user requirements cover functionalities not only required by specific implementation of the pilots.

The users of the ZDMP pilots provide not only a valued input for the requirements but also the necessary feedback for the fine tuning of the requirements.

The purpose is therefore to obtain a clear definition of the set of ZDMP functionalities that are sufficient and feasible which results in a requirements collection.

At the same time the project’s scope and boundaries are defined (ie what is in-scope and what is out-scope),

It was necessary to have available a large spectrum of expertise and multi-disciplinary well-balanced teams of experts to make it possible to obtain multiple, alternative, and heterogeneous visions of the needs and project scope leading to the project requirements.

This is why, the main categories of the ZDMP consortium members are:

- User partners from different industrial sectors who provide valuable insights into the needs of the real industry. Noting that special attention must be paid not to design functionalities which are dedicated only to narrow application domains.
- Technical partners have considerable experience supporting the technology industry and are always updated on the latest trends, which can help with the analysis of the market alternatives, existing services, platforms, technologies, and methodologies.

1.3 Methodology to Produce the Deliverable

This task is a joint effort between all ZDMP partners involved. The initial phase of this task was aligned with the starting activities of the ZDMP project. These activities were about selecting the pilot projects and gathering the knowledge and expertise of all partners. Common face-to-face meetings on the industrial partners premises, via interviews, via remote teleconference, and (even more remotely) with the use of sets of with the pilots' industrial partners took place. From this joint effort, two deliverables are considered: the ZDMP (D2.3) - Industry Scenarios and Use Cases and the Requirements Document and Update (D4.1ab).
For the Use Cases and the Requirements Document deliverable, the ZDMP team defined a methodology for the whole requirements development process. This process included several steps as described in the sections below:

- Perform an analysis of the project inputs
- Identify the requirements sources
- Determine the type of elicitation to perform for each requirement type
- Define the traceability on the requirements
- Define post-processing actions over the elicited requirements
- Define requirements' change management rules

By using this methodology, the ZDMP requirements life cycle follows the project life cycle, and encompasses the steps of:

- Elicitation of requirements
- Requirements specification validation from stakeholders
- Design with traceability towards requirements
- Development of the functionalities that fulfil the requirements
- Evaluating the functionalities, with traceability between test cases and requirements
- Test results communication to stakeholders
- Acceptance of requirements implementation from stakeholders

1.3.1 Inputs Analysis

The first stage regarding the preparation for the requirements elicitation was to clearly identify the relevant inputs including that from stakeholders. Besides identifying the stakeholders, there was a need to determine a representative of each stakeholder - ie someone that can help to understand the stakeholder views and who can validate the finalised requirements.

The identification of the relevant inputs thus considered separate views of the project, provided by:

- **ZDMP Users**: The “customers” identified more clearly were the ZDMP pilots, as one of the objectives of the project is to develop zApps that aim to satisfy the needs of these manufacturing companies:
    - **Pilot 1.1:** MRHS + Ford, represented by MRHS
    - **Pilot 1.2:** MRHS + Ford, represented by Ford
    - **Pilot 1.3:** MRHS + Ford, represented by Ford
    - **Pilot 2.1:** HSD + FIDIA + FORM represented by FORM
    - **Pilot 2.2:** HSD + FIDIA + FORM represented by FORM
    - **Pilot 2.3:** FIDIA + FORM represented by FORM
    - **Pilot 3.1:** ALFA + CONT represented by CONT
    - **Pilot 3.2:** CONT represented by CONT
    - **Pilot 3.3:** MASS + CONT represented by CONT
    - **Pilot 4.1:** PTM + FLEX represented by FLEX
    - **Pilot 4.2:** CEI + ALONG represented by ALONG
    - **Pilot 4.3:** FLEX+ ALONG + CONS represented by CONS
    - **Pilot 4.4:** FLEX+ ALONG + CONS represented by CONS
- **ZDMP Mission and Vision.** Important inputs for the development of the project were the ZDMP Mission and Vision. It is essential that the project would not resume to simply being a solution to fit the needs of the pilots; it must have its own objectives and vision. It is crucial that the project must fit the needs to many others industrial users, not only the needs of the industrial pilots. Hence, these major stakeholders need to care about:
    - ZDMP ideas, objectives, purpose, and mission
    - What the ZDMP consortium wants for future exploitation purposes
    - How to achieve the desired step beyond the state of the art
    - Technological limitations
    - Budget limitations
    - The definition of what is in and out of the scope of ZDMP
    - Business policies
    - Other constraints (eg project limitations, EC regulations etc)
- **ZDL Viewpoint.** Whilst ZDL is not yet formed, and indeed the ZDMP Mission and Vision strongly represents the ZDL concepts, its specific ‘current’ viewpoint would be an important input for the development of the requirements. As well as its own role and viewpoint, ICE also best represents ZDL as editor and lead of D1.1 (Vision) and D3.5.2 (ZDL). However, as ZDL is established and is running, the Managing Director of ZDL will be requested to review all requirements to ensure that they are compatible with running the ZDL business.

The pilots and other parties mentioned above, will be responsible for providing inputs regarding the specific needs related to their own scope, to defend their views and positions, and to validate the completeness and sufficiency of the elicited requirements.

1.3.2 Definition of Requirements Sources

The elicitation of requirements described in detail in Section 2 was performed taking into account multiple activities and methods which included:

- Interviews (description of the job done by the customer, in their own words).
- Apprenticing (the developer becomes apprenticed to the user and accompanies the user to learn the job by observation and asking questions).
- Use cases documentation analysis
- Observing structure and patterns of the work.
- Existing software and equipment analysis
- Brainstorming

1.3.3 Definition of Requirement Levels and Traceability

Requirements should be:

- Clear: Can be understood in the same way by all the persons involved with a minimum of additional explanations
- Concise: Are explained in a straightforward way in order to be understood with a minimum effort
- Verifiable: There can be performed some tests or other testing approaches, like inspection or demonstration, in order to determine if the product is implementing adequately every request
- Consistent: Do not interfere with other requests
- Complete: Every request has to fully describe the functionality which has to be delivered

As the requirements are gathered, they are evaluated and clarified to:

- Identify measurable requirements
- Identify conflicting requirements, which should be explained to the customer. Some requirements can be refused to avoid conflicts
- Classification of requirements into:
    - Functional
    - Non-functional (performance, operational, maintainability, security, log, and audit, “look and feel”, constraint, political, availability, etc.)

When performing the requirements elicitation, the team defined the type of each requirement that was being elicited, which is intrinsically tied to the elicitation phase (see Section 2.1). These types reflect some of the characteristics that are associated to the requirement itself, such as:

- **User/Customer/Business Requirements** gathering from each of the thirteen pilots is performed using business analysis techniques. These include the analysis of concepts including business processes, roles with responsibilities, rules, events, and business entities. They form the High-Level User Needs Specification and are specified as “User Requirements” in the Requirements system
- **System Requirements** refers to the identification and specification of ZDMP technical requirements of the solution related to working context and environment of the solution, technical architecture and standards, user interfaces, and other requirements for core users and custodians of the system. Even if they will be derived from the previously defined User Requirements, some of these requirements may be new and not related to the business requirements. These will be the most valuable requirements for supporting the definition of the ZDMP Architecture (WP4)
- **Software Requirements** refers to all technical specificities that derive from the User and System requirements. Where system requirements roughly represented high-

level modules, software requirements detail this specification, using mainly software developer language not limited to:

- Identification and specification of solution system data assets
- Identification and specification of solution data flows and important data source and data destinations in the system
- Identification and specification of important solution application, security, integration services with a full set of their characteristics
- Identification and specification of solution information architecture:
    - Decomposed to main modules: Application systems, services, databases, and their mutual relationships
    - Architectural design of future system, focused (but not limited) to automation, integrated data environment, user services, security
    - Emphasize on unified and seamless integration with other external systems for data exchange
- Identification and specification of logical/business concepts introduced and/or exploited in the solution

They represent a low-level analysis down to the detail of implementation of the functionalities and constraints that are needed for the thirteen pilots, specified in a language that is close to the software developers. These form the foundation for the ZDMP software development tasks.

These levels of requirements should evolve with a proper traceability, where each higher-level requirement should trace to at least one requirement of the next lower level. There are two types of exceptions:

- Some requirements of a lower level cannot be traced upwards to a higher level because they are specific to that level of detail and it is not appropriate to “invent” a higher-level requirement just for that.
- Some higher-level details are already specified in a manner where they can be interpreted as lower-level requirements, and it is useless to have duplicate requirements just to maintain the traceability.

1.3.4 Best practices for Requirements Post-processing

Eliciting the requirements is only the first step; the entire process needs to be defined regarding the correct storage, maintenance, and change management for each requirement during their lifetime. This includes:

- Establishing unambiguous terminology: The ZDMP project glossary will be used for clarifying the ambiguities found in the requirements terminology
- Developing the requirements: This will be not only for the actual development of the list of requirements but also enhancements of existing requirements for a better definition
- Identifying inconsistencies, duplicates, and contradictory requirements: After the list of requirements is captured, there is the need to analyse it and search for potential problems
- Analysing and validating the requirements: The final list is then compiled and validated by the respective user. This means the users have an opportunity at each elicitation phase to analyse the requirements text, checking if they agree with it and if it is compliant to their pilots’ constraints and needs

- Managing changes: All changes to any requirement must be performed with the approval of its accountable person and changes must be marked with a version change for each changed requirement
- Impact analysis: For all changes an impact analysis must be performed from the user's point of view as well for the consequences for the ZDMP eco-system
- This will be made available in an easy way since the requirements are also an on-line living deliverable: www.zdmp.eu/requirements

1.4 Literature analysis

To achieve the results on this deliverable, the team made use of several best practices, mostly from:

- Software Engineering Institute/Carnegie-Mellon University's CMMI® [CKS11]
- Project Management Institute's Body of Knowledge PMBoK® [PMI13]
- European Space Agency's Guide for Software Requirements Definition [ESA95]
- Papers and research in the area [DFi03, DFi07, IHo93]

1.5 Main Results and Outputs

The main results and outputs of this document are specified on the list of requirements stated in Annex C, which includes:

- The ZDMP functional requirements

2 Elicitation Activities

The main activity of the requirements development process is about eliciting the requirements. For this purpose, the methodology defined in Section 1.3 was used to obtain the specification of requirements, using activities such as user and technical meetings, direct and indirect observation, analysis of documentation, business, user needs, and the existing products on the market.

Several meetings were conducted to establish a consensus view and to point out the conflicts in defining and measuring functionalities. These meetings helped the customers and elicitation team to express their views on ZDMP.

2.1 Strategy

The strategy that was taken for the elicitation was defined during the kick-off meeting of the project, and included the definition of five “customer” pilots that were responsible for providing their vision of the ZDMP project, as can be seen in Figure 1.

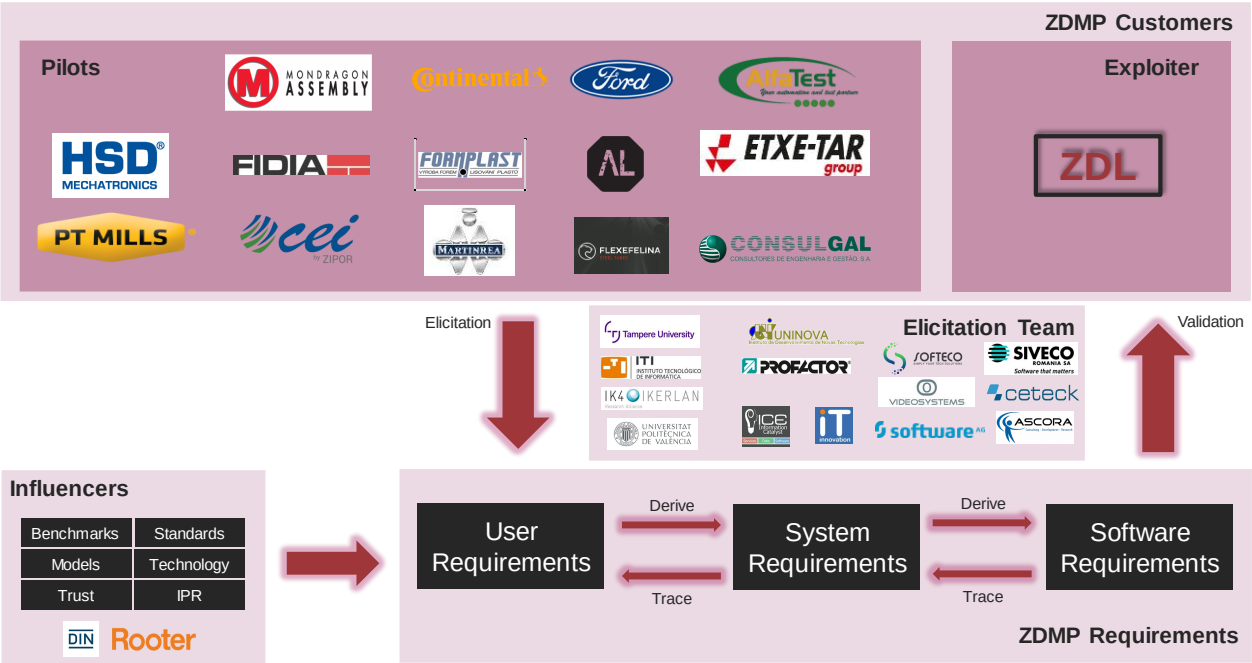


Figure 1: Elicitation Team

2.1.1 User Requirements

The elicitation of the User Requirements was performed starting from D2.3. A list of requirements was generated based on the described use cases by teams of two analysts. The requirements were accompanied by questions addressed to the users and technological partners. The requirements were then updated and submitted for analysis to the users and technological partners. Following their analysis through face to face meetings or telecoms involving representatives from multiple partners, a consolidated version of user requirements was released (Annex C). This approach allowed people from multiple partners to become more familiar with each other, with the intention to help in future project tasks.

2.1.2 System requirements

The same approach is taken for the system requirements elicitation phase and following phases. Multi-disciplinary teams organize either focus groups or brainstorming sessions individually for each pilot in order to be able to add enough details to the user requirements. In this way it is possible to get the system requirements from the User Requirements. Analysts from all partners will contribute, generating a set of system requirements from the list of User Requirements for each pilot. The system requirements have to be validated by the “customers”. The validation process could then induce more iterations, as necessary.

2.1.3 Software requirements

Finally, from the list of User and System Requirements, technological partners will be asked to generate the software requirements for each software component from the pilots. Then a walk-through of all pilots by all partners is performed. The resulting set of requirements can then be validated by the “customers” thus generating a set of new software requirements. The validation process could also induce further iterations.

2.2 Guidelines to Perform the Requirements Elicitation

Prior to performing the elicitation, a set of rules and guidelines to help the teams to perform their work was established in order to achieve the best results possible with the best use of the available resources.

2.2.1 Analysis of the Scope of Requirement Types

The analysis of a business needs to consider multiple disciplines and views over that business, in order to be able to correctly capture all the needs that are required. These views on the business include [ESA95] the following major types of requirements:

- **Functional Requirements:** A function is a 'defined objective or characteristic action of a system or component' and a functional requirement 'specifies a function that a system or system component must be able to perform'
- **Non-Functional Requirements:** Are those requirements that do not directly describe a functionality of the system, but instead describe concepts which are related to characteristics of that system. These include:
    - **Performance Requirements**: Numerical values for measurable variables used to define a function (eg response time, rate, frequency, capacity, speed, or accuracy)
    - **Interface requirements:** Hardware, software, or database elements that the system (or system component) must interact or communicate with
    - **Operational requirements**: How the system will run (ie when it is to be operated) and how it will communicate with human operators. Operational requirements may describe physical aspects of the user interface. Descriptions of the dialogue, screen layouts, and command language styles are all types of operational requirements
    - **Resource requirements**: The upper limits on physical resources such as processing power, main memory, disk space etc. They may describe any requirement that the development or operational environment place upon the software. A resource requirement should state the facts about the resources, and not constrain how they are deployed

- **Verification requirements**: Constrain the design of the product. They may do this by requiring features that facilitate verification of system functions, or by saying how the product is to be verified
- **Security requirements**: Requirement for securing the system against threats to confidentiality, integrity, and availability. They should describe the level and frequency of access allowed to authorised users of the software. If prevention against unauthorised use is required, the type of unauthorised user should be described
- **Portability requirements**: How easy it should be to move the software from one environment to another. Possible computer and operating systems, other than those of the target system, should be stated
- **Quality requirements**: The attributes of the software that make it fit for its purpose. The major quality attributes of reliability, maintainability, and safety should always be stated separately. Where appropriate, software quality attributes should be specified in measurable terms (ie with the use of metrics)
- **Software Reliability, Availability and Serviceability**: The ability of a system or component to perform its required functions under stated conditions for a specified period of time even in stress conditions and hardware components failures
- **Safety requirements:** Specify any requirements to reduce the possibility of damage that can follow from software failure. Safety requirements may identify critical functions whose failure may be hazardous to people or property
- **Backup and recovery requirements:** Specify any requirements to reduce the possibility of data loss that can follow from hardware and/or software failure
- **Communication requirements:** Specify any requirements related to hardware and software infrastructure needed to assure the data change between different information systems

2.2.2 Requirements Elicitation Best-Practices

In order for the ZDMP platform and components to be effectively designed, implemented, and measured, the requirements must be specific, unambiguous, and clear. To achieve that, the elicitation teams got numerous directions regarding how to act, including:

- Write simple, clear, and unambiguous statements
- Apply proper language according to the requirements level
- Testable (verification evidence should be stated)
- State what ZDMP should have and what it should NOT have (rejected requirements)
- State dependencies between requirements (different from traceability)
- Apply versioning
- State source (see Section 1.3.2)
- Prioritise - According to MoSCoW method, requirements have different degrees of prioritization. A requirement ranking is delineated by its keyword:
    - "**Must have**" is used to define mandatory requirements.
    - "**Should have**" is used to define important requirements
    - "**Could have**" is used to define desirable requirements
    - "**Won't have**" is used to define requirements that will be integrated in the future

A set of presentations were held right before each elicitation session, first in F2F meetings, then via e-mail. These presentations tended to show to the eliciting teams the purpose of

each eliciting phase and how to do it, presenting “good” examples on “how to do it” and “bad” examples on “how not to do it”. Namely, a strong concern from the start was how to harmonise the structure of a requirements statement. As requirements tend to get complex and subject to interpretations, the idea was to define at least a standard structure for requirement statement containing the:

- Actor: Who is required to do the action
- Priority: Mandatory requirements
- Action: The action which has to be performed by the Actor. The action may consist of more than one closely related action joined by “and”, but not by “or”.
- What: Subject of the action
- Why: Reason of the action

This is the formal requirement statement structure considered for all requirements that will be elicited.

2.3 Requirements Scope

The definition of the scope of the ZDMP framework will be determined by the list defined in Annex C as being according to the PMBoK the features and functions that characterize a product, service, or result. Hence, only the explicitly stated items are considered in-scope and everything else is considered out-of-scope.

Additionally, even from this specified list, which is a living document, the field “Requirement Specification Status” determines the scope analysis performed over the requirement specification. This may have the values:

- **Proposed**: States that someone proposed the requirement statement, but it has not been validated by the user
- **Validated**: States that the user agrees and validated the statement
- **Rejected**: States that the user did not agree with the statement and is either subject to change or will be dropped.

Thus, all requirements that will not have this field with the value “Validated” or “Proposed” will be considered out-of-scope.

Another analysis that needed to be performed is whether a requirement will be feasible according to the vision of ZDMP and the effort, cost, and other constraints of the project implementation itself. This will be performed by creating a field named “Requirement Implementation Status”, which will be the result of the analysis performed during the project by the ZDMP team over the requirement.

This field may have the values:

- **Proposed**: States that the requirement is accepted by the user and thus proposed for implementation, but it has not been yet validated by the ZDMP implementation team
- **Accepted**: States that the requirement was accepted for implementation, but it has not been implemented yet
- **Rejected**: States that although the user wanted it, the ZDMP implementation team considers that the requirement is out of the scope of the project and it will not be implemented. The decision will be based on the Cost-Benefit Analysis. This decision has to be agreed by the user and it will be considered out of scope once this is agreed by the user

- **Being implemented**: States that the requirement was accepted and is already started to be implemented but it is not fulfilled yet
- **Implemented**: States that the requirement was accepted and was completely implemented, however it has not been yet validated by the customer
- **Validated:** States that the customer agreed and validated the implementation of the requirement

2.4 Analysis and Post-Processing

After the elicitation is performed, an analysis is performed over the large set of requirements for the ZDMP framework. The idea is to analyse all requirements, looking for redundant information, ambiguous statements, erroneous statements, and contradictory requirements.

As the requirements are elicited per use case and zApp, a special attention is given to requirements which involve the same zApp for more use cases. In this situation the requirements are tuned to cover all the use cases.

There are also performance requirements which have to be carefully analysed in order to minimize the development costs. It makes no sense to impose performance requirements (eg response times) which are not really needed in real situations.

Every sector organized combined groups, formed by users, technical leads from WP9/10, and additional partners to allocate WP5-8 Tasks/Components (as defined in the D4.3 Architecture document) or to allocate/confirm that any requirement is more specific and would be addressed in the Pilot WPs 9/10.

Then, all technical partners were requested to go through each requirement in their role as specific component(s) developer and a general partner perspective to verify:

- To validate a requirement allocated to their component – ie to either accept it or reject it and identifying why it is inappropriate
- To examine if any requirement should be allocated to their component

Upon completion of the above, the requirements and technical manager then addressed any anomalies including the stimulation of multi/bilateral discussion to ensure they were addressed.

3 Requirements Specification

The output of the requirements elicitation process after the post-processing consists in a spreadsheet similar to the one documented in Annex C.

It is a major concern of the team to make the requirements as standardized as possible in the same format. Section 2.2.2 targets the structure of each requirement statement (ie its text), this section describes the structure of the whole requirement record. This includes, besides the requirement statement, a set of fields which are needed for providing additional information to the requirement. The meaning of each of the fields (ie columns) that can be found on this spreadsheet is:

- **Requirement Definition:**
    - **Use Case:** Specifies the use case which owns the request
    - **Actor ie Component:** The element of the ZDMP framework which owns the requirement
    - **RID:** Requirement identifier: RQ_<xxxx> (egRQ_0001) where xxxx is a sequential number
    - **Source:** Document from which the requirement is elicited
    - **Use Case or other document Quote:** Quoted text describing the functionalities from which the requirement was elicited
    - **Generic/Specific:** The requirement has a general coverage or is a specific requirement for the module being analysed – in general generic requirements are likely to be addressed in WP5-8 and specific requirements in the zApp building in WP9-10
    - **Requirements Definition:** Description of the requirement
    - **Task/Component:** For each requirement it is specified which task/component(s) from the ZDMP platform is responsible
    - **Questions to answer:** Questions resulting from the requirement analysis to be addressed by users or by technical partners
    - **Current answers:** Answers to the questions
    - **Comments:** Made during reviews, face to face meetings, call conferences or brainstorming meetings which may aid clarification
    - **Dependencies**: Determines if there are any dependencies for this requirement (eg other requirements connected or images or documents)
    - **Platform or Application:** The requirement targets the ZDMP platform or a specific zApp
    - **Phase**: Which can be “User Requirements”, “System Requirements”, or “Software Requirements”
    - **Pilot**: This field may include information about the applicability of the requirement regarding which of the thirteen defined pilots effectively originated the requirement
- **Priorities:**
    - **ZDMP Priority (DOA)**: The priority as perceived from the commitments in the DOA (Provided by Siveco as document owner)
    - **Customer Priority (Pilots)**: The priority as specified from the users. Each sector was asked to roughly allocate according to the following distribution scheme to avoid the users (unrealistically but invariably!) allocating all to ‘must’

    - **40%: "Must have**" is used to define mandatory requirements.
    - **30%:** "**Should have**" is used to define important requirements
    - **30%:** "**Could have**" is used to define desirable requirements

  - **Customer Priority (ZDL)**: The priority as specified by ZDL. Since ZDL does not exist yet, this is only an initial allocation made by ICE as temporary owner through D3.5.1. This is also according to the 40/30/30 "rule" above
  - **Technical Priority (ZDL)**: The priority as specified by relevant technical partners. This is necessary since some feature may be necessary to implement but invisible to users
  - **Combined Priority (Implementation)**: This is the final 'communally agreed' opinion based on the aforementioned priorities. It is not a calculation as such, but a rough assessment. Once again, the 40/30/30 allocation "rule" was used

- **Implementation information:**
  - **Lead:** The company that is responsible for addressing this requirement if it is in the Must/Should category. This is related to the role tables of the architecture
  - **Version:** If a requirement changes an indicator for version control. For M9 they are all set at "1"
  - **Parent requirement -** Requirement identifier of the requirement from which the current requirement derived,
  - **Requirement Specification Status**: Determines the status of the elicited requirement, which can be:
    - **Proposed**: Initial proposal for requirement
    - **Validated**: Requirement elicited proposal for functionality was validated by the "customer" (see Section 2.3)
  - **Requirement Implementation Status**: Determines the status of the elicited requirement with respect to its implementation in the ZDMP system, which can be:
    - **Rejected**: Requirement will not be implemented
    - **Accepted**: Requirement will be considered for implementation
    - **Being Implemented**: Requirement is in the process of being implemented
    - **Implemented**: Requirement has been implemented
    - **Validated**: Requirement implemented functionality was validated by the "customer"
  - **Notes:** Allowance for further comments

4 Conclusions

This task extracts requirements from the end users to formulate technical needs to be implemented. Also, this exercise allocates which task/component would fulfil each requirement. The requirements will be used as input to the specifications defined in WP4 (Architecture) in order to determine the core set of features needs by ZDMP. In addition, it clarifies how the architecture will interface with identified external systems and meet the expected performance criteria. From the perspective of the forthcoming deliverables, this will define needs, and specify how functionalities are expected to work, and whom to contact in case developers could not understand certain functionality.

Furthermore, there will be a significant input from these requirements to the use case specification and implementation deliverables in WP4-WP8, as the User Requirements provide a good indication of which functionality should be implemented. These work packages apply and validate ZDMP by implementing the tools, services, and middleware developed by the consortium over specific and real-world scenarios.

Subsequently, the Requirements Spreadsheet will be used to record the validation results for each requirement, particularly those defined as Essential for the customer, which implies ZDMP must deliver the function. As such, the living version of the Requirements Spreadsheet (Annex C)will act as a dynamic document that will be updated throughout the life of the project and can provide focus on development, implementation, validation efforts and will be the basis for the deliverable D4.1b in M48.

Annex A: History

Document History	
Versions	V0.1: • First contents framework based on vf-OS process V0.2: • Tuning of process to reflect needs of ZDMP V0.3: • Generation of first pass of XSL framework and aligning document V0.4: • Extraction of requirements by entire team in workshop sessions in Calpe Plenary V0.5: • Shaping XLS from comments/requirements to date and interim work V0.6: • Results from Bremen plenary sector/technical sessions and component owner walk-throughs V0.7: • Final edits of document and start of prioritisation activity V0.8: • Addressing comments after Review 1, Ronal Bejarano (TAU) V0.9: • Addressing comments after Review 2, Marc Dorchain (SAG) V1.0.0: • Addressing comments after PM Review and M9 Submission V1.0.1: • Addressing comments after M9 Review and resubmission including the enclosure of 'final' M9 requriements XLS
Contributions	SIVECO: • Radu Lazar, Majority of document and first requirements extraction from use cases as well as overall coordination of requirements repository. Assessment of DOA/Combined priority and cross checking of information. ICE: • Stuart Campbell, Requirements framework, multiple document sections and initial ZDL priority. Also, interim, and final reviews of both document and requirements system. ITI: • Santiago Cáceres Elvira Technical Priority and resolving technical anomalies All: • Practically the whole ZDMP team participated in eliciting, refining, and confirming requirements

Annex B: References

None

Annex C: ZDMP Requirements

For the purposes of the EU submission the list of ZDMP Requirements is available as a Attached Excel spreadsheet file with the name “EU-ID D32 - ZDMP-ID D4.1a - Requirements Document (M9) - Annex v1.0.1 (Resubmission).xlsx. However, this should be considered as a ‘snapshot’ and the ‘real’ version will be a living document maintained throughout the project’s life and where changes and status will be documented. Another final snapshot of this will be taken at M48 and accompany the Requirements Document and Update (M48). This will also enable the evaluation of which requirements could and could not be implement and will also make input to ZDL.

This separation was performed so that the readers may experience an improved handling over the requirements, being able to perform actions over them such as sorting and filtering.

The live version of the Excel is available as http://www.zdmp.eu/requirements. The document is marked as public and available to all.

ZERO DEFECTS
Manufacturing
Platform

ZDMP

www.zdmp.eu