

Project information

Project Acronym:	DRIVER+
Project Full Title:	Driving Innovation in Crisis Management for European Resilience
Grant Agreement:	607798
Project Duration:	72 months (May 2014 - April 2020)
Project Technical Coordinator:	TNO
Contact:	coordination@projectdriver.eu

Deliverable information

Deliverable Status:	Final
Deliverable Title:	D932.11 – Functional design of the PoS database
Deliverable Nature:	Report (R)
Dissemination Level:	Public (PU)
Due Date:	November 2017 (M43)
Submission Date:	26/01/2018
Sub-Project (SP):	SP93 - Solutions
Work Package (WP):	WP932 – Portfolio of Solution design
Deliverable Leader:	ARMINES
Reviewers:	Hector Naranjo, GMV Peter Petiet, TNO Marcel van Berlo, TNO Pedro Soria, ATOS Denis Havlik, AIT Tim Stelkens-Kobsch, DLR
File Name:	DRIVER+_D932.11_Functional design of the PoS database.pdf

DISCLAIMER

The opinion stated in this report reflects the opinion of the authors and not the opinion of the European Commission.

*All intellectual property rights are owned by the DRIVER+ consortium members and are protected by the applicable laws. Except where otherwise specified, all document contents are: “©DRIVER+ Project - All rights reserved”.
Reproduction is not authorised without prior written agreement.*

The commercial use of any information contained in this document may require a license from the owner of that information.

All DRIVER+ consortium members are also committed to publish accurate and up to date information and take the greatest care to do so. However, the DRIVER+ consortium members cannot accept liability for any inaccuracies or omissions nor do they accept liability for any direct, indirect, special, consequential or other losses or damages of any kind arising out of the use of this information.

Revision Table

Issue	Date	Comment	Author
V0.1	08/10/2017	Initial draft	Sébastien Truptil, ARMINES WP932 Leader
V0.11	02/11/2017	Improvements and extensions throughout the document	Denis Havlik, AIT
V0.12	02/11/2017	Integrating significant new requirements from various partners (integration by Sébastien Truptil, contributors are listed as authors)	Rosa Viera, ATOS Jaime Martin, ATOS Chiara Fonio, JRC Emil Wrzosek, SRC Todor Tagarev, CSDM Stine Bergerson, PRIO Alice Clemenceau, Valabre Andre de Rond, SRH Laurent Dubost, TCS Wiktor Gawroński, SGSP Camilo Palacio, ARC Adam Widera, WWU Thomas Obritzhauser, FRQ
V0.2	27/11/2017	Updated version based on internal review	Sébastien Truptil, ARMINES Aurélie Congès, ARMINES
V0.3	30/11/2017	Updated version based on internal review	Sébastien Truptil, ARMINES Aurélie Congès, ARMINES
V1.0	04/12/2017	Final check and submission	Francisco Gala, ATOS
V1.1	15/12/2017	Updates following request from TC. Improved executive summary, other	Denis Havlik, AIT Sébastien Truptil, ARMINES
V1.2	15/01/2018	Updated version following review. Improved part 3	Sébastien Truptil, ARMINES
V1.3	22/01/2018	Updated version following review	Sébastien Truptil, ARMINES
V1.4	23/01/2018	Full review of document	Marcel van Berlo, Technical Coordinator, TNO
V1.5	25/01/2018	Final check and approval for submission	Tim Stelkens-Kobsch, Quality Manager, DLR
V1.6	25/01/2018	Final check and approval for submission	Peter Petiet, Project Director, TNO
V2.0	26/01/2018	Submission to the EC	Francisco Gala, ATOS

The DRIVER+ project

Current and future challenges due to increasingly severe consequences of natural disasters and terrorist threats require the development and uptake of innovative solutions that are addressing the operational needs of practitioners dealing with Crisis Management. DRIVER+ (Driving Innovation in Crisis Management for European Resilience) is a FP7 Crisis Management demonstration project aiming at improving the way capability development and innovation management is tackled. DRIVER+ has three main objectives:

1. Develop a pan-European Test-bed for Crisis Management capability development:
 - Develop a common guidance methodology and tool (supporting Trials and the gathering of lessons learned).
 - Develop an infrastructure to create relevant environments, for enabling the trialling of new solutions and to explore and share Crisis Management capabilities.
 - Run Trials in order to assess the value of solutions addressing specific needs using guidance and infrastructure.
 - Ensure the sustainability of the pan-European Test-bed.
2. Develop a well-balanced comprehensive Portfolio of Crisis Management Solutions:
 - Facilitate the usage of the Portfolio of Solutions.
 - Ensure the sustainability of the Portfolio of Tools.
3. Facilitate a shared understanding of Crisis Management across Europe:
 - Establish a common background.
 - Cooperate with external partners in joint Trials.
 - Disseminate project results.

In order to achieve these objectives, five sub-projects (SPs) have been established. **SP91 Project Management** is devoted to consortium level project management, and it is also in charge of the alignment of DRIVER+ with external initiatives on crisis management for the benefit of DRIVER+ and its stakeholders. In DRIVER+, all activities related to Societal Impact Assessment (from the former SP8 and SP9) are part of SP91 as well. **SP92 Test-bed** will deliver a guidance methodology and guidance tool supporting the design, conduct and analysis of Trials and will develop a reference implementation of the Test-bed. It will also create the scenario simulation capability to support execution of the Trials. **SP93 Solutions** will deliver the Portfolio of Solutions which is a database driven web site that documents all the available DRIVER+ solutions, as well as solutions from external organisations. Adapting solutions to fit the needs addressed in Trials will be done in SP93. **SP94 Trials** will organize four series of Trials as well as the final demo. **SP95 Impact, Engagement and Sustainability**, is in charge of communication and dissemination, and also addresses issues related to improving sustainability, market aspects of solutions, and standardization.

The DRIVER+ Trials and the Final Demonstration will benefit from the DRIVER+ Test-bed, providing the technological infrastructure, the necessary supporting methodology and adequate support tools to prepare, conduct and evaluate the Trials. All results from the Trials will be stored and made available in the Portfolio of Solutions, being a central platform to present innovative solutions from consortium partners and third parties and to share experiences and best practices with respect to their application. In order to enhance the current European cooperation framework within the Crisis Management domain and to facilitate a shared understanding of Crisis Management across Europe, DRIVER+ will carry out a wide range of activities, whose most important will be to build and structure a dedicated Community of Practice in Crisis Management, thereby connecting and fostering the exchange on lessons learnt and best practices between Crisis Management practitioners as well as technological solution providers.

Executive summary

This deliverable D932.11 “Functional Design of the PoS Database” is part of SP93 and in particular WP932 (PoS design). It defines the initial functional design of the Portfolio of Solutions Database (PoS DB) early in the project¹. The DRIVER+ Trial plan includes four Trials, the first one being planned for June 2018 with a dry run in March 2018. During each of those Trials, lessons learned and potentially more requirements will be collected. These will be implemented in the following versions of the PoS DB until the end of the project. This means that a first version of the PoS DB is needed before Trial 1, therefore this document is an initial version. The deliverable provides input to WP933, where the updated user expectations will be managed in an agile co-creation process throughout the project. This document does not address the actual technical design and implementation, as this is a task of WP933 and will be reported in D933.11 “DRIVER+ online tools – implementation specification”. Moreover, an updated list of functionalities will be provided in deliverable D932.12 “PoS tutorial and recommendations”.

The deliverable is a result of the interaction of the PoS design team with different PoS DB users. This interaction includes a workshop session and discussions at the DRIVER+ kick-off meeting and additional discussions at a joint SP92/93 meeting in November 2017 supported by weekly PoS design teleconferences in October and November 2017. Furthermore, it includes responses to a PoS requirement survey filled out by several project members to gather their opinion about the functionalities and the users that the PoS DB should address. Finally, it also includes direct inputs by key DRIVER+ partners, WP and SP leaders as well as the feedback gathered from selected DRIVER+ Crisis Management practitioner partners (on graphic user interface and provided functionality) through usage sessions of the first available prototypes.

This deliverable describes the initial set of the required PoS DB functionalities, processes and data models as well as the type of users that will use the PoS DB. These functionalities are essentially linked to a Content Management System (CMS) that provides access rights, provides editorial workflows and research contents. In addition, it also reflects on the possible implementation.

The primary function of the PoS DB is to document all the relevant available solutions by DRIVER+ consortium partners, as well as solutions from other external organizations in such a way that different stakeholders can easily use this information. However, the DoW clearly indicates that the PoS DB is more than just a simple searchable database of solutions and that it must provide added value to all the DRIVER+ project members as well as to the external stakeholders. In summary, these are the main guiding principles that the PoS DB should embrace:

- Contribute towards a shared understanding of crisis management across Europe.
- Facilitate the design, execution, monitoring and gathering of the lessons learned for/from Trials.
- Facilitate the description of tools, solutions, Test-bed elements and the relations between them.
- Propose a “marketplace” where Trial owners can discover relevant solutions for their Trials.
- Store all the results of Trials: the addressed gaps, the type of practitioner organizations that have used it, the regulatory conditions that apply, societal impact considerations, etc.

One of the requirements is that the PoS DB should be easy to use. In particular, the following often-used functions should be optimized for the best possible user experience:

- The search mechanism should be simple and finding the relevant information should not take too much of the user’s time. One of the consequences of this high-level requirement is that the PoS data should be tagged by a single (known and used) taxonomy.
- Adding new content should be possible without an extensive training and the actual entering of the data should be straightforward.

¹ We are referring to DRIVER+ timeline here, with September 2017 being the project (re)start.

Table of Content

The DRIVER+ project.....	4
Executive summary	5
1. Introduction	10
1.1 Deliverable structure.....	13
2. Methodology.....	14
3. Collection of requirements	16
3.1 Needs identification	16
3.1.1 Survey	16
3.1.2 Meetings and weekly teleconferences.....	18
3.1.3 Joint SP92/SP93 meeting.....	20
3.2 Requirements of the Portfolio of Solutions	20
4. Information treatment system design	22
4.1 User types of the Portfolio of Solutions and their expectations.....	22
4.2 Functional specification of the Portfolio of Solutions database	23
4.3 Relationship between the functionalities of the Portfolio of Solutions	31
4.4 Processes between users.....	32
4.4.1 Content validation process.....	33
4.4.2 Solution integration validation process.....	35
4.4.3 Support of guidance methodology.....	36
4.5 Data model of the Portfolio of Solutions	36
5. Information system design.....	39
5.1 Selected Drupal plug-in for the Portfolio of Solutions	39
5.2 Hierarchy of the Portfolio of Solutions website.....	43
6. Conclusion	45
References.....	46
Annexes	47
Annex 1 – DRIVER+ Terminology	47
Annex 2 – User participation by activities	49
Annex 3 – Octopus diagrams to define PoS functionalities.....	52

List of Figures

Figure 1.1: Key functions of the PoS and the Guidance Tool	10
Figure 1.2: Overview of the relations between the main PoS DB design elements.....	11
Figure 1.3: Information System (IS) definition	12
Figure 1.4: Model, View, Controller (MVC) architecture	12
Figure 2.1: Overview of the methodology used.....	14
Figure 4.1. Generic overview of the octopus diagram	23
Figure 4.2. Octopus diagram for Trial stakeholders	24
Figure 4.3: Dependency between functionalities	32
Figure 4.4. Overview of the functionalities by priority and urgency.....	32
Figure 4.5: Content validation process.....	34
Figure 4.6: Solution integration validation process.....	35
Figure 4.7: Initial proposed data model of the PoS DB	38
Figure 5.1: Hierarchy of the Portfolio of Solutions Database website	44
Figure A1: Octopus diagram for Trial stakeholders.....	52
Figure A2: Octopus diagram for solution providers	52
Figure A3: Octopus diagram for editorial board	53
Figure A4: Octopus diagram for tool providers.....	53

List of Tables

Table 3.1: Answers to the survey	17
Table 3.2. Collected requirements and feedback in one on one meetings and teleconferences.....	18
Table 3.3. Collected requirements and feedback during the joint SP92/SP93 meeting	20
Table 3.4. Functional requirements of the Portfolio of Solutions Database (PoS DB).....	21
Table 3.5. Non-functional requirements on the Portfolio of Solutions Database (PoS DB).....	21
Table 4.1: PoS DB user types and their expectations.....	22
Table 4.2: List of expected generic functionalities of the PoS DB.....	24
Table 4.3: List of expected specific functionalities of the PoS DB.....	27
Table 5.1: Drupal plug-in selected for the PoS DB	40
Table A1: DRIVER+ Terminology.....	47
Table A2: List of participants	49

List of Acronyms

Acronym	Definition
API	Application Programming Interface
BPMN	Business Process Model and Notation
CM	Crisis Management
CMS	Content Management System
DoW	Description of Work
EB	Editorial Board
IS	Information System
ITS	Information Treatment System
KoM	Kick off Meeting
MVC	Model View Controller
PoS	Portfolio of Solution
PoS DB	Portfolio of Solution Database
REST	Representational State Transfer
SIA	Societal Impact Assessment
SOAP	Simple Object Access Protocol
SP	Sub-Project
TAP	Trial Action Plan
TRL	Technology Readiness Level
WP	Work Package
US	User Story

1. Introduction

Crisis Management (CM) is a complex process where a multitude of CM Functions² has to be performed by the relevant actors in a timely and satisfactory manner, in order to minimize the adverse effects of crisis situations. In order to do so, CM organizations need to achieve and provide the matching CM capabilities. When an organization cannot match all the necessary CM Functions with the required capabilities, it can be said that there is a “CM Gap”. CM Gaps can be either organization-specific, in a sense that satisfactory CM Solutions for the problem at hand already exist on the market but for some reason have not been or cannot be deployed by that organization; or generally valid for all CM organizations, because no satisfactory solutions have been developed yet.

In DRIVER+, the CM Solutions are defined as a set of predefined methods for addressing specific CM Functions/Gaps that can be realized or supported by CM Tool(s). In this context, CM Tools can be either software or hardware and CM Solutions can combine technologies, tools, methods, concepts, or recommendations that regard potential technical, organizational, procedural, legal, policy, societal, or ethical improvements to the European Crisis Management legacy. This also means that one tool or method could be used in several solutions.

This deliverable provides an initial functional design of the “Portfolio of Solutions Database” (PoS DB). The PoS DB aims to provide a shared place where stakeholders meet around a solution. To a lesser extent, the deliverable also suggests some elements of the initial design of the “Guidance Tool”³, which is based on the Guidance Methodology developed in SP92. The PoS DB documents the practitioners’ needs and the available solutions, whereas the Guidance Tool addresses the design, execution, and assessment of the Trials. This can be best understood by looking at the holistic picture that is presented in Figure 1.1.

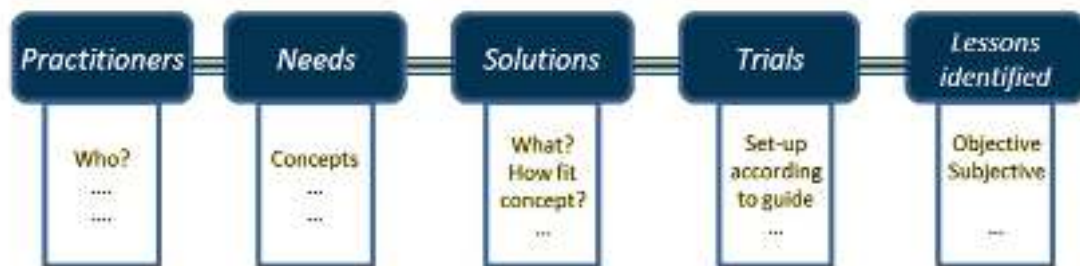


Figure 1.1: Key functions of the PoS and the Guidance Tool

In this figure, the PoS DB covers mainly the left-hand side (practitioners, needs, and solutions), whereas the Trial Guidance Tool addresses the right-hand side (Trials and lessons identified). This differentiation is easily understood at an abstract level and it is having implications for the detailed design of both tools (e.g. the elements provided by one tool are often re-used in the other). They will, therefore, be realized either as a single service or as two closely interconnected services.

As a part of this work, a better understanding for the key elements of the PoS DB design has been developed, including the main user types and the data elements these users may contribute to or need. Figure 1.2 summarizes the relations between these elements. On the one hand, either practitioners (e.g. CM professionals) are looking for solutions that achieve CM Functions or they want to evaluate the adequacy of solutions and CM functions during a Trial. On the other hand, solution providers aim at

² An extensive taxonomy of the CM functions has been provided in the deliverable D934.10 - Taxonomy of CM functions for classification of solutions.

³ “Trial Guidance tool” is a sister site of the PoS that focuses on the task of defining, executing and assessing the Trials. The intention is to support these two functions by a single web site.

developing their solution(s) based on tools and methods proposed by tool providers and method providers. They also want to have their solution(s) tested during Trials.

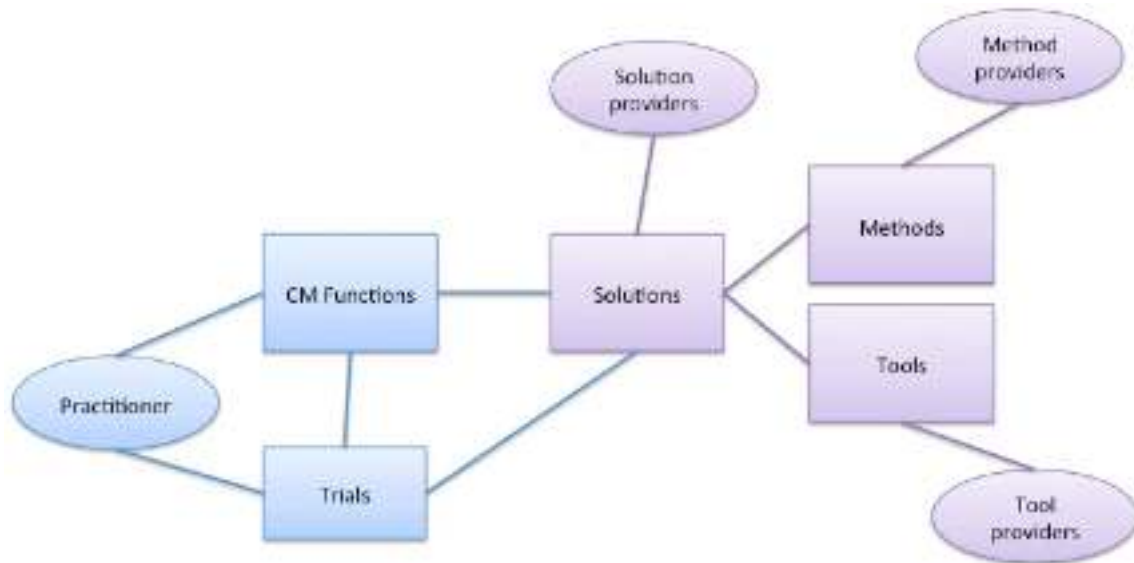


Figure 1.2: Overview of the relations between the main PoS DB design elements

PoS DB is an Information System (IS) as defined by Bernus, Peter, Nemes, Laszlo et Williams (1) and Saadoun (2). Regarding the IS definition of Morley (3) and Morley et al. (4), an IS can be divided into two systems, represented by Figure 1.3:

1. **Information Treatment System (ITS):** independent of informatics realization that is in charge of the processing of information.
2. **Informatics System (IS):** a constellation of software used in order to support the processing of information.

This vision of an IS underlines the fact that an IS is not limited to the software although nowadays, information systems could not be imagined without an informatics system, especially because of the search functionality.

To ensure the coherence between the ITS and the IS, it is necessary to first define the ITS. Benaben (5) explains that it is characterized by a process, which is a sequence of activities in order to achieve at least one objective. Therefore this process is composed of activities, each one of them transforming certain inputs into certain outputs. Information about these inputs and outputs needs to be stored and retrieved. It is also describing the three layers of the ITS (Process, Activity, Information) which can be translated to a computational point of view (Workflow, Software, Data Model). Figure 1.3 summarizes the layers of an IS and the mapping between an ITS and an IS.

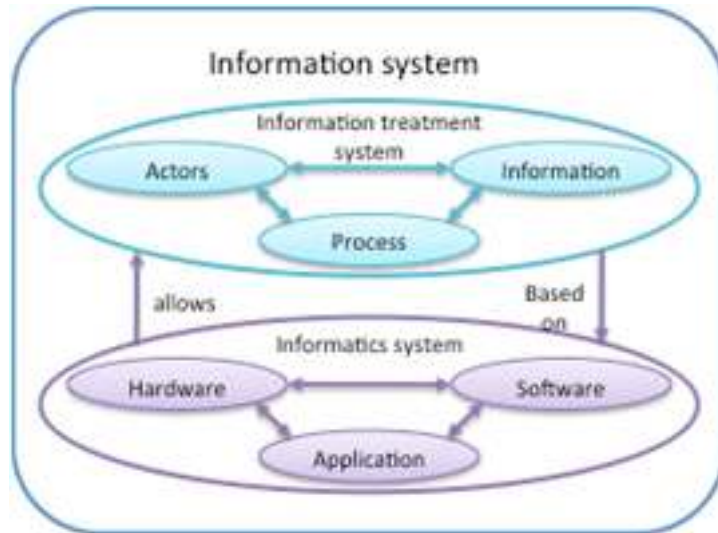


Figure 1.3: Information System (IS) definition

From the IS perspective the PoS DB will be designed as a website. One of the design decisions is that it will be developed based on a Model, View, Controller (MVC) software architectural pattern, which is a common architecture for a website. As illustrated in Figure 1.4, MVC architecture is composed of three elements:

- Model: Independent of the user interface, aiming at managing the data.
- Model: Independent of the user interface, aiming at managing the data.
- View: aims at showing contents of the model to users.
- Controller: manages the interaction between model, view, and user.

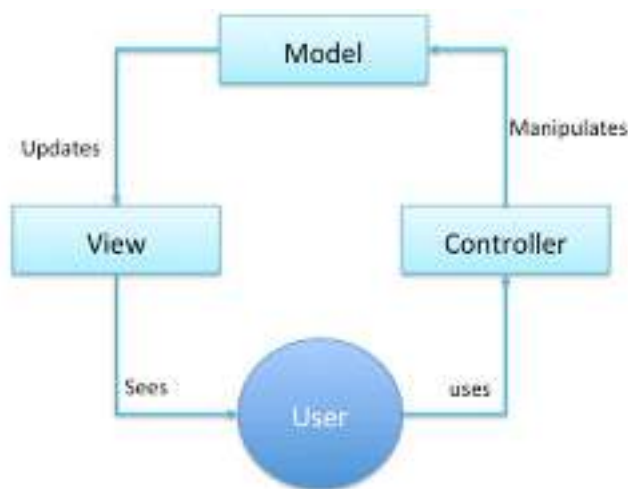


Figure 1.4: Model, View, Controller (MVC) architecture

1.1 Deliverable structure

Based on the definition of the information system divided into the ITS and the IS, this deliverable aims at describing the functional design of the PoS database. The system functionality (i.e. ITS) is firstly defined independently from its implementation (i.e. IS) and then the proposed implementation(s) are being described.

Therefore, the deliverable is organized as follows:

- Section 2 describes the methodology used in order to define the PoS DB functionalities.
- Section 3 describes the users' expectations and the activities performed in order to collect them.
- Section 4 describes the ITS point of view of the PoS DB.
- Section 5 describes the IS point of view of the PoS DB and the implementation choices for the PoS DB realization, in anticipation of the D933.11 DRIVER+ online tools – implementation specifications.
- Section 6 provides the conclusions and way forward.

2. Methodology

The methodology used in order to define the functional design of the PoS DB is inspired by Zehtaban and Roller (6) and represented in Figure 2.1. This methodology is based on an iterative approach: all along the development of the PoS DB, new requirements or functionalities will probably be requested. Indeed, DRIVER+ Trial plan includes four Trials, the first one being planned for June 2018 with a dry run in March 2018. This means that a first version of the PoS DB is needed before Trial 1. During each of those Trials, lessons learned and potentially more requirements will be collected. These will be implemented in the following versions of the PoS DB until the end of the project, which is why this document is just an initial version. It is therefore needed to have an initial functional design leading to a first version of the PoS DB before Trial 1, which will be modified according to lessons learned during each Trial. Therefore, the methodology used covers WP932 (functional design of the PoS DB) and WP933 (Implementation of the PoS) activities.

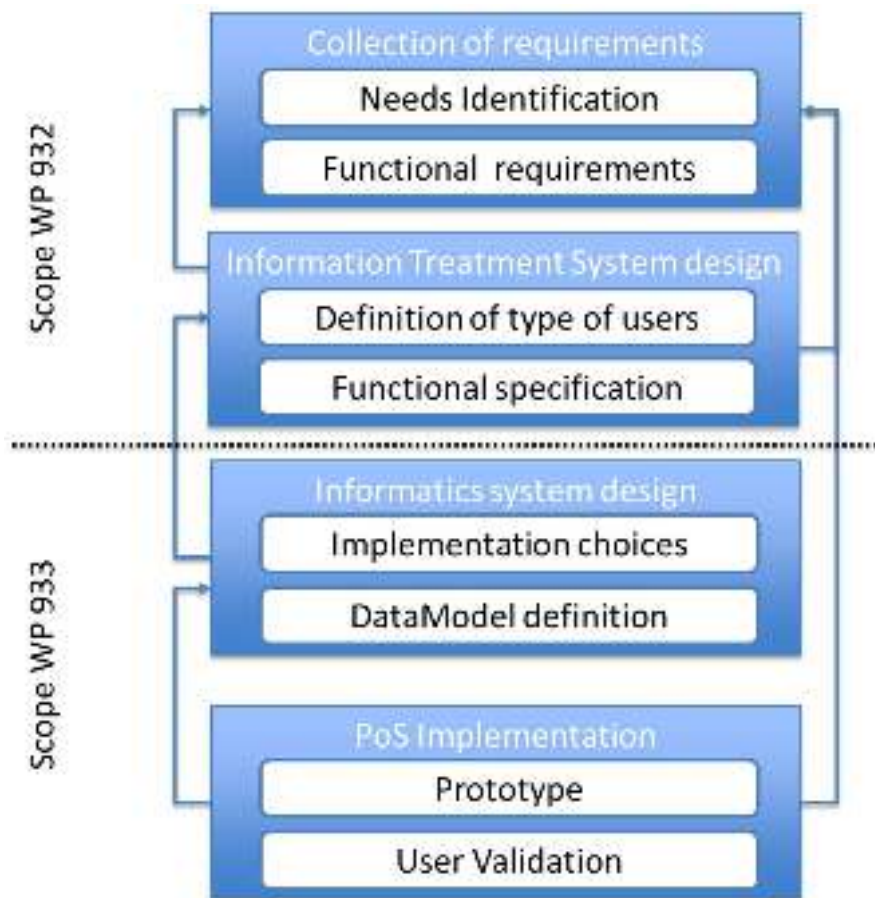


Figure 2.1: Overview of the methodology used

The methodology used is divided into four steps (each step is described in detail in the following section):

1. Step 1: Collection of requirements

This step aims at gathering information from users in order to define the PoS DB functional requirements. These requirements must be functional and non-functional requirements.

2. Step 2: Information Treatment System design

Based on PoS DB functional requirements, this step aims at describing the ITS of the PoS DB first by identifying the functional specification of the PoS DB and the design specification and secondly defining the different type of users and their specific expectations.

These two first steps are within the scope of this deliverable. Even if this deliverable introduces next steps, they will be described in detail in deliverable “D933.11 - DRIVER+ online tools – implementation specifications”.

3. Step 3: Informatics System design

This step aims at describing the implementation choices and the defined data model.

4. Step 4: PoS Implementation

This step aims to develop the PoS DB and improve it based on the feedback from users.

If necessary, the design process could be restarted at any step of the process.

3. Collection of requirements

3.1 Needs identification

The requirements that are listed in this deliverable had to be gathered in a very short period following the DRIVER+ re-start since a first version of the PoS DB has to be implemented before the first Trial. Moreover, they had to be gathered from a large variety of the users with different expectations and also aligned with the requirements that are already written down in the DoW. With this in mind, a combination of methods to identify the needs was applied, namely distributing a survey, enabling providing direct inputs into the deliverable, frequent discussions during face to face meetings and teleconferences and individual discussions with the key stakeholders.

Therefore, in order to better understand the specific needs of the different types of the PoS stakeholders and establish an initial functional design of the PoS DB, the following activities have been performed:

1. Discuss the PoS DB design with all DRIVER+ partners during the project restructuring phase. The results of this process are already included, as initial ideas for the PoS DB design, in the DRIVER+ DoW.
2. Present the initial PoS DB design ideas at the DRIVER+ kick-off meeting in September 2017 and discuss them with the DRIVER+ partners that participated in this event.
3. Prepare a generic survey that has allowed all DRIVER+ team members to provide any PoS DB requirements that they have, analyze the answers and cluster and summarize the inputs. This process and the results of this survey are described in section 3.1.1.
4. Discuss the requirements and functional design for the PoS DB and the Guidance Tool at 12 dedicated SP93 and joint SP92/SP93 teleconferences from September to November 2017. This process is described in section 3.1.2.
5. To enhance the information provided by the survey results, additional user requirements have been requested and received by means of providing direct inputs to this deliverable. With this step, significant inputs from 31 DRIVER+ project members with different profiles (ranging from the practitioners and Trial owners, and Trial methodology, community and business development experts, to solution owners) have been received. The users who actively participated in preparing this deliverable are listed in the deliverable changelog as well as in Annex 2 of this deliverable.
6. The initial PoS DB prototype was presented at the joint SP92/SP93 meeting held in Ispra in November 2017 and further steps towards supporting the development and execution of the Trials were discussed. This process is described in section 3.1.3.

In summary, the PoS DB has to be built in a sustainable way and answer to the expectations listed in section 3.2 and detailed in specific functionalities in section 4.2.

The following section will describe these activities in more detail.

3.1.1 Survey

The aim of the survey is to gather information from potential users of the PoS DB. Moreover, the survey has to contain an open question in order to avoid guiding the answers. Therefore the survey is based on the five whys technique (7). This technique was developed by Sakichi Toyoda for the Toyota Industries and it consists of asking oneself up to five times why you need something or why you have this problem to solve.

This approach could be regarded as simple, but according to Serrat (7), it ensures:

- Accurate and complete statements of problems or requirements.
- Complete honesty in answering the questions.
- The determination to get to the “real” requirements or problems.

The last point is extremely important because each participant has the tendency of proposing solutions rather than expressing their requirements. Thanks to the five whys technique, it is possible to collect the requirements as well as the proposed solutions.

Table 3.1 shows an overview of the collected answers grouped by type of users.

Table 3.1: Answers to the survey

Type of users	Expectations
End-user / Practitioner	In the PoS database, we want to find solutions for their gaps and needs can be found in the PoS database.
	It must be easy to select solutions in different categories.
	Provide a way to test a solution or see the result of previous Trials. Comparative table of solutions for a same gap/function Select solutions thanks to research fields (TRL, function, gaps, previous Trial, ...) Find feedback on solution (opinion of other End-users).
	Find solution as model, not only software. Request solution for a gap/need.
	Models of natural risks' behaviour (fire, flooding). Strong solution for telecommunication (air/air; air/ground; ground/ground). COP tools (information exchanges, cartography, handrail, means' management table). Innovative sensors. Do some researches on the following subjects: CM solutions provided by research projects from TRL7. Know where and by whom the solution is used and the users' opinion. Contact of solution provider. Tutorial for the use of the solution. Input/output of the solution (type, format). A table comparing several solutions for a given domain. Solution's price. Norms' references to the solution.
Researchers	Provide a list of research gaps, a place to share in order to innovate. Propose solution in order to fill the gap.
	The PoS is showing solutions categorized on several levels, i.e. not only per technical functionality but also on a more overarching level, understandable also to non-technical users (e.g. for research purposes, such as "drones", which can then be sub-categorized as "data collection" or similar). It would also be nice to have a clear definition of what a solution is, in the first meeting with the PoS DB. A well-functioning glossary also seems very important. In the former DRIVER project, a framework for societal impact assessments was developed, in which the functions of the tools and measures were assessed. In the DRIVER+ project, the DRIVER+ solutions against an already defined set of societal impact criteria will be assessed. There already is a first version of this (a taxonomy of the functions), but that also needs updating based on the current (and future) solutions in the PoS. Matching the functions (on at least one of the levels: "drone" or "data collection", as mentioned above) that the PoS DB operates with, with the functions that the Societal Impact Assessment (SIA) framework is based on would be mutually beneficial. Furthermore, training modules (something like a booklet or a practical "SIA toolkit") on how to take societal impact into account in European CM will be provided by WP913 by

Type of users	Expectations
	the end of the project. One possibility could be to integrate this into the PoS DB in some way. However, this needs to be further discussed with the SP leader.
Trials Stakeholders	<p>Descriptions providing a clear and complete understanding of solutions functionalities. What it does, how, where is the innovation? Efficient way to find a solution that completes your need (without having to read and analyse every solution description in PoS)</p>
	<p>Descriptions providing a clear and complete understanding of solutions functionalities. What it does, how, where is the innovation? Efficient way to find a solution that completes your need (without having to read and analyse every solution description in PoS). Practitioners need to know exactly in what they’re going to invest their time. If the description lacks some important info, is too general or poorly written, they will not be interested. And having clear descriptions has also benefits for the Trial organizers. Practitioners don’t have too much time to browse through databases. In previous projects’ field events there were situations when practitioners were approached with a laptop or tablet and they were shown the solutions. In less than 5-10 minutes they lost their interest. The good first impression needs to be secured - by ensuring the quality of description (completeness, comprehensibility, understandability). Sometimes they just randomly choose and read one or two solutions from PoS - if the description was poor, they were not interested in further participation in project.</p>
Solution providers	<p>Description of the solutions must be efficient. Advertise for their solution. Manage several versions of a solution and its component. Could apply to Trials. Could describe solution through user stories.</p>
All	Provide a commonly shared glossary. List of gaps, list of functionalities.

3.1.2 Meetings and weekly teleconferences

To improve the gathered feedback, it was decided to organize individual meetings with different users of each type. A dozen one on one meetings were organized to talk about what the participants needed and what they would like to see in the PoS DB. Table 3.2 summarizes the results of those meetings.

Table 3.2. Collected requirements and feedback in one on one meetings and teleconferences

Type of user	Expectations
Trial stakeholders	<ul style="list-style-type: none"> • Single field for purpose and scope in the Trial. • Paragraph type for overall research request, with at least a title and a description of each research request. • One short chapter for general information of scenario. • It must be possible to clone a Trial in order to keep track of the different versions and make the edition easy. • Trial must be connected to taxonomy. • Tab with problem description and unfindable tabs for the CM functions. • The guidance tool should follow the structure of the Trial Action Plan (TAP). • The logic for creating a Trial must be easy to understand. • The general information must be detailed in order to be as useful as possible to

Type of user	Expectations
	<p>solution providers.</p> <ul style="list-style-type: none"> • A timeline with dates might be a good way to visualize the Trial.
Solution providers	<ul style="list-style-type: none"> • Organization of the PoS DB in chapters, with different tabs organizing the different parts of a solution. • Presentation of the tools used in a solution, and link to the detailed presentation of those tools. • Adding more tools to a solution after the creation of the solution must be possible. • Description fields are needed, such as general information (purpose, crisis type, and size, CM functions, gaps addressed...), tools involved, test cases, etc. • Possibility to clarify who is the provider of the solution, what is needed to use it, the needed licenses, etc. • There must be a matching between a solution and Trials. • Solution owners must have a way to answer a call for participation in a Trial. • Solutions may be grouped and categorized to make it easy to find one. • It must be possible to select several types of crisis addressed, or even to select all types (the same applies for the size of the crisis). • Solution providers might want to add some pdf files or other uploaded material. • Be able to clarify what phase of the crisis is addressed by the solution.
Tool providers	<ul style="list-style-type: none"> • Description fields. • Hardware requirements, licenses. • What is the form of the tool (mobile phone application, group of people, software, etc.).
End-users	<ul style="list-style-type: none"> • It must be very clear and easy to understand what to put in what field and how to fill it correctly. • The interface shall be user-friendly. • It should be possible to subscribe to a part of the PoS DB and be notified by email if something changed.
All	<ul style="list-style-type: none"> • The creation of a solution, a tool or a Trial must be intuitive and easy to follow (process step by step). • The difference between what is a solution and what is a tool must be very clear. • Explain the technical terms so that every user can understand it (ex: TRL). • It would be nice to have a presentation of DRIVER+ and a clarification that the project is about crisis management on the first page of the PoS DB. • Some pictures would make the PoS DB more user-friendly than a website with just text. • A search function easy to find and use should be available. • It must be clear who can use the PoS DB, who has an interest in using it, and how it should be used. • Some kind of summary on the first page to say “on this page you will find this...” may be interesting. • The words in the PoS DB should refer to the taxonomy (e.g. links or windows appearing when the mouse hovers over the word).

Type of user	Expectations
	<ul style="list-style-type: none"> • There must be a possibility to add a contact person (for Trials, tools, and solutions). • Each user must be allowed to access just some parts of the PoS DB, not all of them (allow different access right depending on who is using the PoS DB).

3.1.3 Joint SP92/SP93 meeting

During the joint SP92/SP93 meeting in Ispra (29/11/2017 to 01/12/2017), the ongoing version of the PoS DB was presented to Trial stakeholders, solution providers, and tool providers in order to gather their feedbacks and the linked requirements. The results of the discussions are captured in Table 3.3.

Table 3.3. Collected requirements and feedback during the joint SP92/SP93 meeting

Type of users	Expectations
All	<ul style="list-style-type: none"> • Authentication system is needed, with different rights for different roles. • There must be a glossary. • There must be a structured data model for Trials, solutions, test cases, tools, etc. • There must be a way to describe the relations between the entities (solution, Trials, CM functions, tools, etc.). • There must be a taxonomy linked to the data. • There must be a way to search for a solution, and there must be some kind of ranking to evaluate how well a solution answers to a Trial. • Some tutorials must be made to help stakeholders add their solution/tool/Trial. • The PoS DB shall offer the possibility to support more than one language. • The PoS DB shall be linked to the Gap Assessment Process. • Gaps and functions must be linked somehow.
Tool stakeholders	<ul style="list-style-type: none"> • The connection to the Test-bed must be as easy as possible technically speaking.
Solution stakeholders	<ul style="list-style-type: none"> • Providing a solution must be as easy as possible. • Communication must be easy between solution providers and Trial owners.
Trial stakeholders	<ul style="list-style-type: none"> • PoS DB has to support the guidance tools. • There must be a way to link solutions and Trials.

3.2 Requirements of the Portfolio of Solutions

Based on the collected inputs as described in the previous sections, Table 3.4 reflects the actual priorities of the future PoS DB users and must be taken into account in the actual development. In order to meet these expectations, it is necessary to define the roles of actors, the useful information and the content management for the PoS DB. This section is describing these facets of the ITS of the PoS DB.

Table 3.4. Functional requirements of the Portfolio of Solutions Database (PoS DB)

ID	Expectations
E1	PoS DB should contribute towards a shared understanding of Crisis Management across Europe.
E2	PoS DB should facilitate the design, execution monitoring and gathering of the lessons learned for/from Trials.
E3	PoS DB should facilitate the description of CM Tools, CM Solutions, Test-bed elements and the relations between these elements.
E4	PoS DB should propose a “marketplace” where Trial owners can search the solutions for their Trials;
E5	PoS DB should store all the results of Trials. This includes the addressed gaps, the solutions that were tested in the Trial, the type of practitioner organizations that have participated in the Trial, the regulatory conditions that apply societal impact considerations, etc.

In addition, two high-level non-functional requirements for the PoS DB were also established (see Table 3.5).

Table 3.5. Non-functional requirements on the Portfolio of Solutions Database (PoS DB)

ID	Non-functional requirements
CR1	Adding new content should be easy to do without an extensive training and the entering of the data should be straightforward and not take too much time.
CR2	The search mechanism should be simple and find the interesting information (e.g. solutions that could be used in the Trials or Trials where solutions could be tested) should be quick. One of the consequences of this high-level requirement is that the PoS data should be tagged by a single (known and used) taxonomy.

The discussion with the DRIVER+ business developers and community experts indicates that these two requirements are highly important for the sustainability of the PoS DB. In particular, the quality of the search and Trial/solution matching functionality is considered a critical factor for the wider acceptance of the PoS DB.

4. Information treatment system design

The aim of this section is to describe the Information Treatment System (ITS) of the Portfolio of Solutions (PoS). The Portfolio of Solutions Database (PoS DB) is defined as a dynamic catalog of available CM Solutions that could be used in crisis management. Eventually, the PoS DB should help practitioners find the right solution to their needs. This task will be facilitated by the description of the solutions as well as by the documentation of the use of solutions in Trials' context. Moreover, the PoS DB could indicate which kind of desired functionalities are not provided by existing solutions and thus provide a sustainable and flexible roadmap for the development of new solutions by companies and/or researchers.

4.1 User types of the Portfolio of Solutions and their expectations

The PoS DB has to be used by several types of users, in order to achieve the expectations included in the previous section, and the PoS DB has to serve as a meeting point for all of them. Table 4.1 summarizes the target users of the PoS DB. Each type of user has different expectations and requires access granted to different parts of the PoS DB. This should be reflected both in the PoS DB design and in the definitions of the user roles and their access rights.

Table 4.1: PoS DB user types and their expectations

Type of user	Main expectation for the PoS DB
Practitioners	Find solutions that answer their needs. Test the solutions in Trials. Provide feedback or evaluations on the use of these solutions in the Trials context.
Trial stakeholders	Describe Trials. Select solutions. Evaluate the quality of solution for Trials. Plan the use of the Test-bed in the Trials.
Solution stakeholders	Describe their solution. Document the way to use their solution in a Trial. Apply for the solution to be used in a Trial. Receive feedback from Trial stakeholders.
Test-bed stakeholders	Document the Test-bed elements so that the solution providers can integrate against it and so that the Trial owners can plan how to use it in the Trials. Receive information on the integration plans and status for all the tools and solutions Receive feedback on the use of Test-bed in Trials.
Editorial Board (EB)	Ensure the quality of the content and the shared understanding of the used terminology.
DRIVER+ project Test-bed stakeholders members	Achieve information about the relations between various work items and of the work progress. Achieve information about performance indicators for each of the Trials, solutions, etc.
DRIVER+ community stakeholders	Share PoS content with the community, e.g., public information on solutions, Trials, Test-bed. Provide the community with a way to contribute own content, e.g. comments and recommendations on Trials or solutions, own solutions etc.

Type of user	Main expectation for the PoS DB
Standardization stakeholders	Contribute to the shared understanding of the CM in Europe. E.g. through standardization of terminology, CM functions, gaps, Guidance Methodology.
All	Facilitate group discussion on specific data items (e.g. solutions, Trials). Facilitate the usage of a shared vocabulary.

4.2 Functional specification of the Portfolio of Solutions database

Based on the expectations described in Section 4.1, the PoS DB has to provide a set of functionalities to the users. In order to define these functionalities, octopus diagrams have been created.

Octopus diagram⁴ is applied after analyzing the user needs and it is used to define the main functionalities in addition to the constraints, e.g. the non-functional requirements. For this, the first step is to investigate the connections between the PoS DB and the external elements (users, solutions, Trials, etc.). These connections are divided into two lists, as illustrated in Figure 4.1.

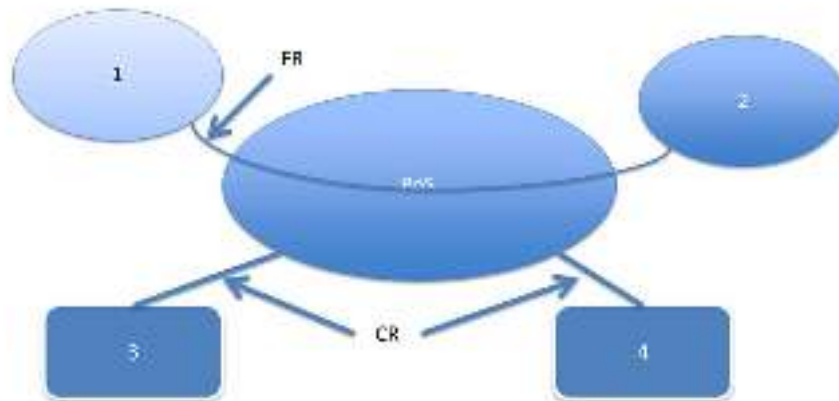


Figure 4.1. Generic overview of the octopus diagram

- Non-Functional / Constraint Requirements (CR): refers to present adoption of the products or why the products will not be used by users.
- Functional Requirements (FR): refers to the interaction of the PoS DB with elements of the surroundings.

The complete set of octopus diagrams can be found in Annex 3. Figure 4.2 illustrates the octopus diagram for Trial stakeholders. Obviously, Trial stakeholders want to have access to solutions, tools, Trials and test cases. All the respective functionalities have to ensure the non-functional requirements (user-friendly way to edit content and simple search mechanism) in order to be accepted by Trial stakeholders. For the sake of readability of Figure 4.2, the generic functionalities (which are required by default) are not represented.

⁴ The Octopus diagram presents the relations between the product and its external environment, through functional and non-functional requirements.

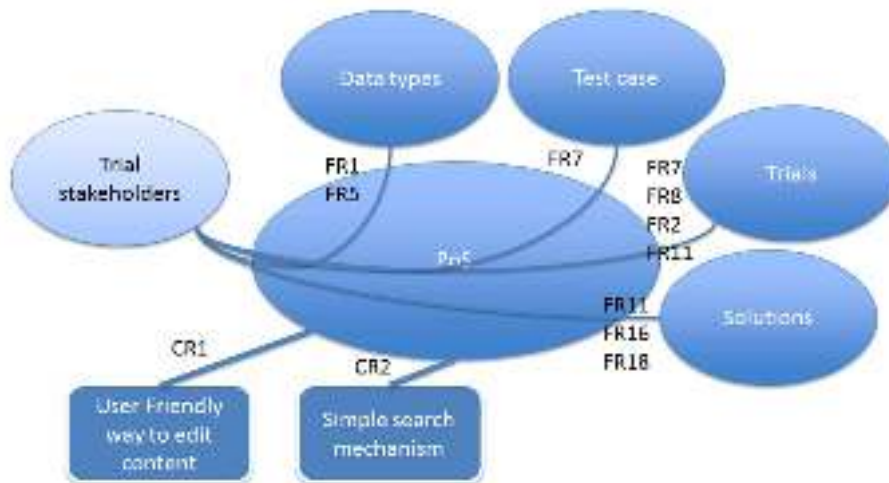


Figure 4.2. Octopus diagram for Trial stakeholders

Based on the definition of the octopus diagram and their analysis, the list of PoS DB functionalities has been defined. Table 4.2 and Table 4.3 provide the expected generic and specific functionalities (respectively) of the PoS, including a priority level (from 5 for mandatory functionality to 1 for optional) as well as including the authorized type of users for such a functionality.

The column “Urgency” indicates how fast a functionality needs to be provided in the project, with 5 being the highest urgency, while the column “Priority” indicates the importance of the feature, with 5 being the most important to be implemented. These levels of urgency and priority have been determined through the analysis of the answers to the survey and the discussions at meetings and teleconferences (see section 3.1). Both the required functions and their urgency and importance should be re-assessed every few months as the PoS DB matures and the users will gain a better understanding of its usage.

Table 4.2: List of expected generic functionalities of the PoS DB

ID	Requirement	Type of User	Description	Priority	Urgency
FG1	Access right	ALL	The PoS DB SHALL ⁵ implement access rights for various data content and for various functions. These access rights should apply to different user roles related to user types in the PoS DB.	5	5
FG2	Tagging of content with taxonomy	ALL	The PoS DB SHALL associate the data with taxonomy terms from one single taxonomy in order to ensure the coherence and the homogeneity of the description. This will allow improving the search and matching capabilities of the site as well as provide this benefit for external stakeholders and standardization bodies.	5	5

⁵ The words “shall”, “must”, “should”, “could”, “may” in this document are to be interpreted as described in RFC 2119 – see <http://www.ietf.org/rfc/rfc2119.txt>.

ID	Requirement	Type of User	Description	Priority	Urgency
FG3	Shared terminology	ALL	The PoS DB SHALL assure that all team members understand and use the terminology in the same way to facilitate the common understanding. This will allow improving the common understanding of the crisis management within the project team and provide benefit to external stakeholders and standardization.	5	5
FG4	Notifications mechanism (e.g. per e-mail)	ALL	In order to assure that the stakeholders who work with the PoS DB are aware of the changes and requests which are of relevance to them, the site MUST provide a way to assign work items to specific team members as well as to “subscribe” to elements of interest, e.g., a solution must have a dedicated “owner” who will be informed of requests for improving the solution description, “editors” who will be informed when these requests are fulfilled.	5	5
FG5	Different views at the same data	ALL	The PoS DB SHALL provide a mechanism to present the data in a form suitable for the needs of the different stakeholders. E.g., the editorial board members and managers will need overviews of the work competition. Trial owners will need Trial-centric views showing which solutions are used in their Trials and the current status of the Trial planning and execution. Solution owners will need a solution-centric view that shows where “their” solution is used, how the users reacted to it, which tools are incorporated, which test cases have been defined for the solution and the status of testing.	5	4
FG6	PoS contents validation	Editorial Board (EB), managers	PoS DB MUST support the editorial process for all data entries, in order to transparently document that the DRIVER+ quality assurance process has been followed and thus the data can be trusted. This is also important for the PoS DB sustainability as the validation makes the PoS information more valuable to the practitioners.	5	3

ID	Requirement	Type of User	Description	Priority	Urgency
FG7	Search by explicitly linked content	ALL	PoS DB MUST allow searching for the data that is explicitly linked to some other content. It SHOULD allow the users to find all the solutions that are used in a specific Trial or to find all the Trials that address specific CM Functions.	5	3
FG8	Search for implicitly associated content	ALL	PoS DB MUST allow searching for the data that is implicitly linked with some other content. It SHOULD allow the users to find all the solutions that may be useful in a specific Trial or to find all the Trials that might make use of a specific solution or to find “similar” Trials and solutions to the one currently viewed.	5	3
FG9	Tasking support	ALL	In order to understand which work items are waiting for a team member to be solved, the PoS DB SHOULD provide some kind of a user-specific “open tasks” list. Such a list needs to show all data items where users action is required, e.g. all the solutions that need to be reviewed by him/her, etc. This mechanism is a kind of TODO list for the PoS DB and complementary to the project management mechanism in place for informing the DRIVER+ team members of new tasks.	4	4
FG10	Cloning of data	ALL	The platform SHALL allow the use of existing content as a basis to develop new content in order to make it easier to describe a new solution or develop a new Trial.	4	4
FG11	Actions and decisions	ALL	To assure that the actions and decisions made by the PoS DB users are traceable and get resolved, some mechanism is needed that will allow any PoS DB user to assign actions to other users, search for specific types of actions and decisions and indicate when such actions/decisions have been resolved, rejected or obsoleted.	4	1
FG12	Search by keywords	ALL	PoS DB SHOULD allow searching for all the data containing a specific keyword.	3	2
FG13	Search by data type	ALL	PoS DB SHOULD allow searching for all the data of a specific type (e.g. CM Solution).	3	2

ID	Requirement	Type of User	Description	Priority	Urgency
FG14	Search by taxonomy tags	ALL	PoS DB SHOULD allow searching for all the data tagged with specific taxonomy terms.	3	2

Table 4.3: List of expected specific functionalities of the PoS DB

ID	Requirement	Type of User	Description	Priority	Urgency
FR1	Dedicated data types for Trials, CM solutions, CM tools, CM functions, CM gaps, solution capabilities and Trial needs / requirements	ALL	The PoS DB SHALL provide dedicated data types for at least these entities: Trials, CM solutions, CM tools, CM functions, CM gaps, solution capabilities and Trial needs and requirements.	5	5
FR2	Select solutions to be used in a Trial	Trial stakeholder and practitioners	The PoS DB SHALL allow the Trial stakeholders to explicitly link a solution already present in the system with a Trial.	5	5
FR3	Describe Test-bed	Test-bed stakeholders	The PoS DB SHALL provide a way to describe the elements of the Test-bed so that the tool providers can plan the integration work and so that the Trial stakeholders can decide which elements of the Test-bed to use in a Trial and how. This COULD be implemented by describing the Test-bed components as a type of solutions. The advantage of this approach is that it allows the same components and solutions to be either tested or used to facilitate testing of other solutions as a part of the Test-bed, depending on the Trial needs.	5	5

ID	Requirement	Type of User	Description	Priority	Urgency
FR4	Describe solution	Solution provider	<p>PoS DB SHALL allow the solution providers to fulfill at least the following metadata on their solutions: solution name, a summary of the solution capabilities, description of the solution “business case”, a detailed description of the capabilities, used tools and methods, responsible party or parties that can support the application of the solution in Trials.</p> <p>PoS DB COULD also facilitate describing the solution training and linking to additional information on the solution, e.g. user manuals, administrator manuals etc.</p> <p>PoS DB SHOULD also allow solution owners and other stakeholders to provide relevant information to use the solution in future Trials.</p>	5	5
FR5	Define mapping and linking relations between these data types	ALL	<p>The PoS DB SHALL facilitate mapping and linking of the data types listed above.</p> <p>The site SHOULD allow the users to explicitly link solutions to Trials, Trial requirements to solution capabilities, and all of these to CM functions and CM Gaps.</p>	5	4
FR6	Describe Tool	Tool provider	<p>PoS DB SHALL allow the tool providers to fulfill at least the following metadata on their tools: Tool name, summary description, Technology Readiness Level (TRL), license terms and other conditions for use, responsible party or parties, etc.</p> <p>It SHOULD also allow the tool providers and other stakeholders to indicate the level of integration of the tool in the DRIVER+ Test-bed.</p>	5	4

ID	Requirement	Type of User	Description	Priority	Urgency
FR7	Describe Trial	Trial stakeholders	<p>The PoS DB SHALL allow Trial stakeholders to provide a description of the Trial.</p> <p>This description MUST provide sufficient information for the solution providers to decide if they could apply for participation in a specific Trial or not.</p> <p>In addition, the PoS DB SHOULD allow the Trial stakeholders to explicitly indicate which solution(s) are used in that Trial, how and what was learned from the Trial for a specific solution.</p>	5	4
FR8	Describe Trial needs and requirements (User Stories)	Trial stakeholder and practitioners	<p>The PoS DB (or the guidance tool) MUST support the Trial stakeholders in describing the detailed Trial needs and requirements on the solutions and also on the Test-bed. This COULD be realized in the form of User Stories and SHOULD facilitate the task of matching the solutions to Trials.</p> <p>The PoS DB COULD allow the solution providers to explicitly link the relevant detailed solution capabilities with the specific Trial needs.</p> <p>Furthermore, the Trial needs SHOULD be linked to CM Functions and CM Gaps and SHOULD facilitate the formulation of the Trial research questions.</p>	5	4

ID	Requirement	Type of User	Description	Priority	Urgency
FR9	Describe solution capabilities (test cases)	Solution provider	<p>The PoS DB SHALL allow the solution providers to include a detailed description of the solution capabilities in form of a verifiable step by step checklist (test case). Test cases COULD be included as separate data items and SHOULD be explicitly linked to the solution on the one and to the Trial on the other.</p> <p>In addition, the solution capabilities SHOULD also be linked to relevant CM functions and gaps. For practical reasons, PoS DB MAY need to differentiate between the inherent solution capabilities that are provided as a part of the solution description and the Trial-specific “offer” that is provided as a response to one or more Trial needs and requirements.</p>	5	4
FR10	Describe the level of integration of a tool or solution in the Test-bed.	Tool and solution providers	<p>The PoS DB SHALL provide a way to indicate how the tool or a solution is or will be integrated into the Test-bed as well as to indicate who and when verified the claims and the results of this verification. For practical reasons, this COULD also be modeled as a Test Case.</p>	5	4
FR11	Help Trial stakeholders to pre-select the solutions for use in a Trial	Trial stakeholder and practitioners	<p>The PoS DB MUST aid the Trial stakeholders in the preselection of solutions. For this, the platform MUST be able to match the Trial needs with the solution capabilities and provide a list of “best matches” to the Trial stakeholders.</p>	5	4
FR12	Help the solution providers to apply for the Trials	Solution providers	<p>The PoS DB SHOULD help the solution providers in finding the appropriate Trial calls for participation and answering to such calls.</p> <p>This COULD be facilitated by mean of the Trial-specific test cases that are explicitly linked with one or more of the Trial user stories.</p>	5	4

ID	Requirement	Type of User	Description	Priority	Urgency
FR13	Validate tool integration	Test-bed	The results of the tests to be performed in order to validate the interaction between the tool and the Test-bed SHALL be included in the PoS DB.	5	4
FR14	Mechanism for maintaining the terminology and taxonomies that are used in the PoS DB	Editorial Board (EB), management	PoS DB SHALL provide a mechanism to assure that term descriptions can be improved and new terms added in an orderly manner, with minimal overhead.	5	3
FR15	Validate test cases	Tool and solution providers, Trial stakeholders, managers	The PoS DB SHALL provide a way to validate the test cases. Trial-specific test cases should be validated through Trials, whereas the integration test cases will be validated through integration testing.	5	3
FR16	Indicate the results of the trialing for a specific solution	Trial stakeholder and practitioners	The PoS DB SHOULD allow the Trial stakeholders to provide the results of the evaluation of a specific Trial or user story for a specific solution. This COULD be done by validating the relevant test cases.	4	4
FR17	Search for CM solutions or CM tools by CM functions	Trial stakeholders, practitioners	PoS DB SHALL provide a comparative table of solutions and tools for certain functionality.	4	2
FR18	Search solution or tools by Trials	Trial stakeholders, practitioners	PoS DB SHALL provide an overview of solutions and tools for a certain Trial.	3	2
FR19	Recommendations system	End-user / Practitioner	PoS DB COULD provide a functionality that will recommend solutions to a user based on his previous navigation or marks (“this Trial seems interesting”) to enhance that practitioners can find solutions in an effective way.	2	2

4.3 Relationship between the functionalities of the Portfolio of Solutions

The octopus diagram does not clarify the relationship between the functionalities because it only expresses the needs of the users. Figure 4.3 underlines the dependencies between functionalities while Figure 4.4 represents the functionalities by priority and urgency of functionalities listed in Table 4.2 and Table 4.3 in section 4.2, required by each type of user which has been described in section 4.1. The blue circle corresponds to specific functionalities while the purple circle corresponds to generic functionalities.

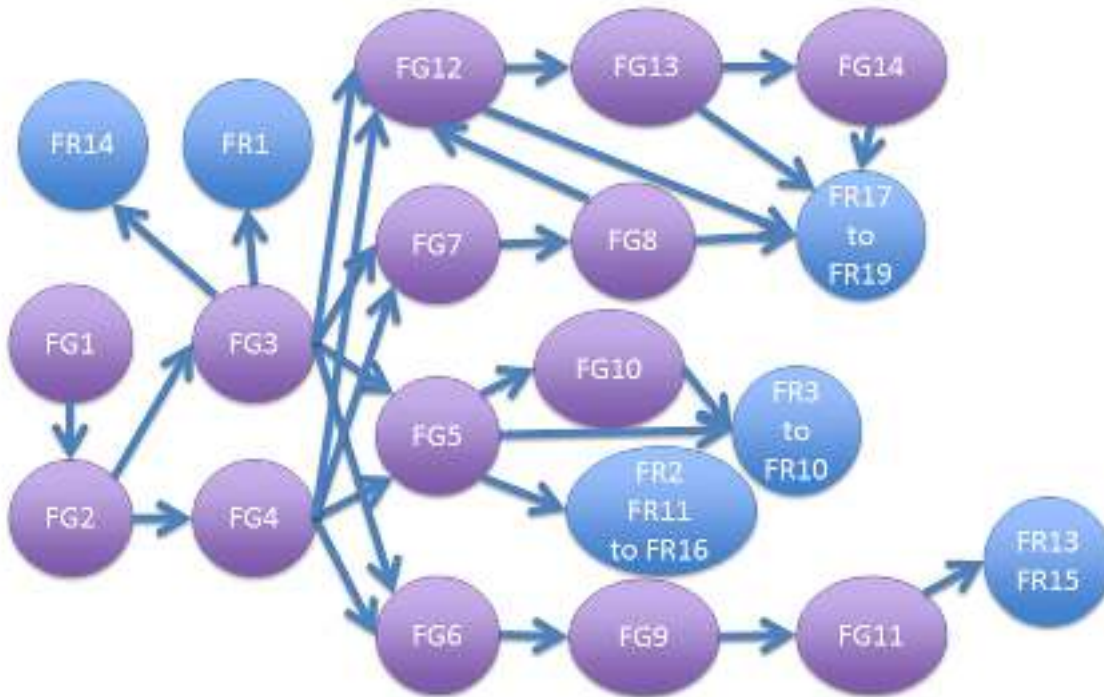


Figure 4.3: Dependency between functionalities

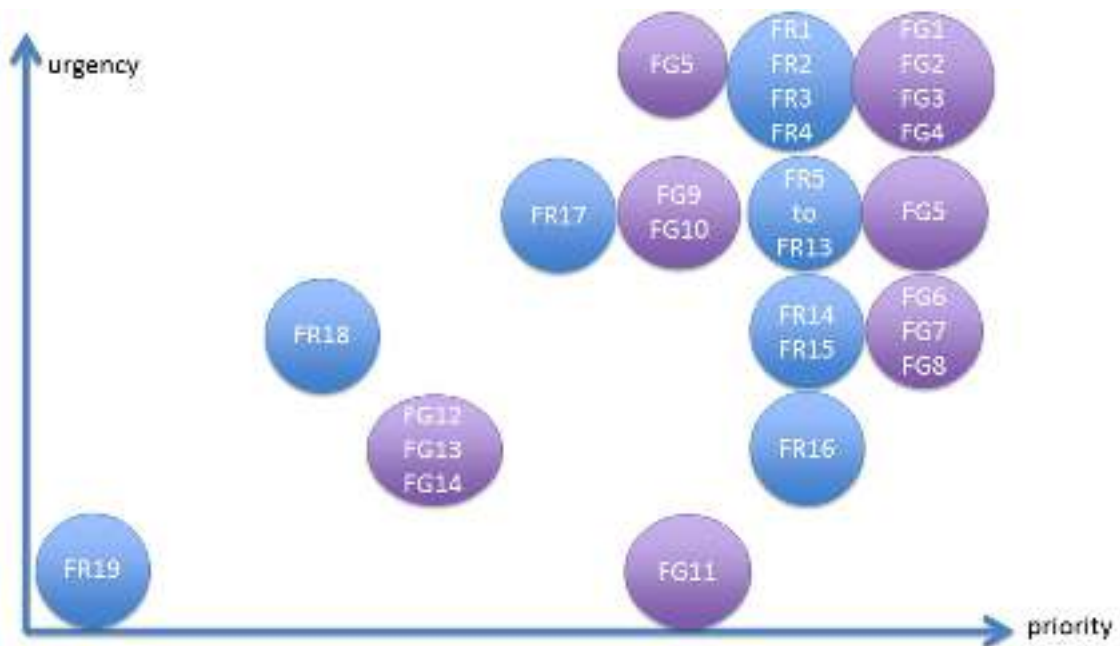


Figure 4.4. Overview of the functionalities by priority and urgency

4.4 Processes between users

When certain functionality requires several users to perform in parallel, it is necessary to define a process that supports and defines the interactions between them. Therefore, this section defines the processes between users of the PoS DB. Processes are described using Business Process Model and Notation (BPMN) as described by OMG (8).

4.4.1 Content validation process

The initial proposal for the PoS content validation process is presented in Figure 4.5. This process aims to ensure the quality of all the new or updated PoS contents. The process is executed through the interaction of two types of users: PoS editorial board users and the content providers (like for instance solution providers and Trial stakeholders).

In order to assure the PoS DB sustainability, the validation process must be designed independently from the DRIVER+ project. For practical reasons during the DRIVER+ project, the PoS editorial board members, nevertheless, need to be nominated by the relevant WP and task leaders, with an explicit acknowledgment of the DRIVER+ quality management.

The process begins with the submission of some new content in the PoS DB. The new content could be an update of a previous description or a brand-new content. If the new content is an update, only the review by one editorial board (EB) user will be needed, otherwise review from two different editorial board users will be required. When reviews are finished, a decision is required. If the quality of the content is deemed sufficient, the PoS DB is updated and notification of the acceptance is sent to the description provider, otherwise, the description provider is informed of the refusal.

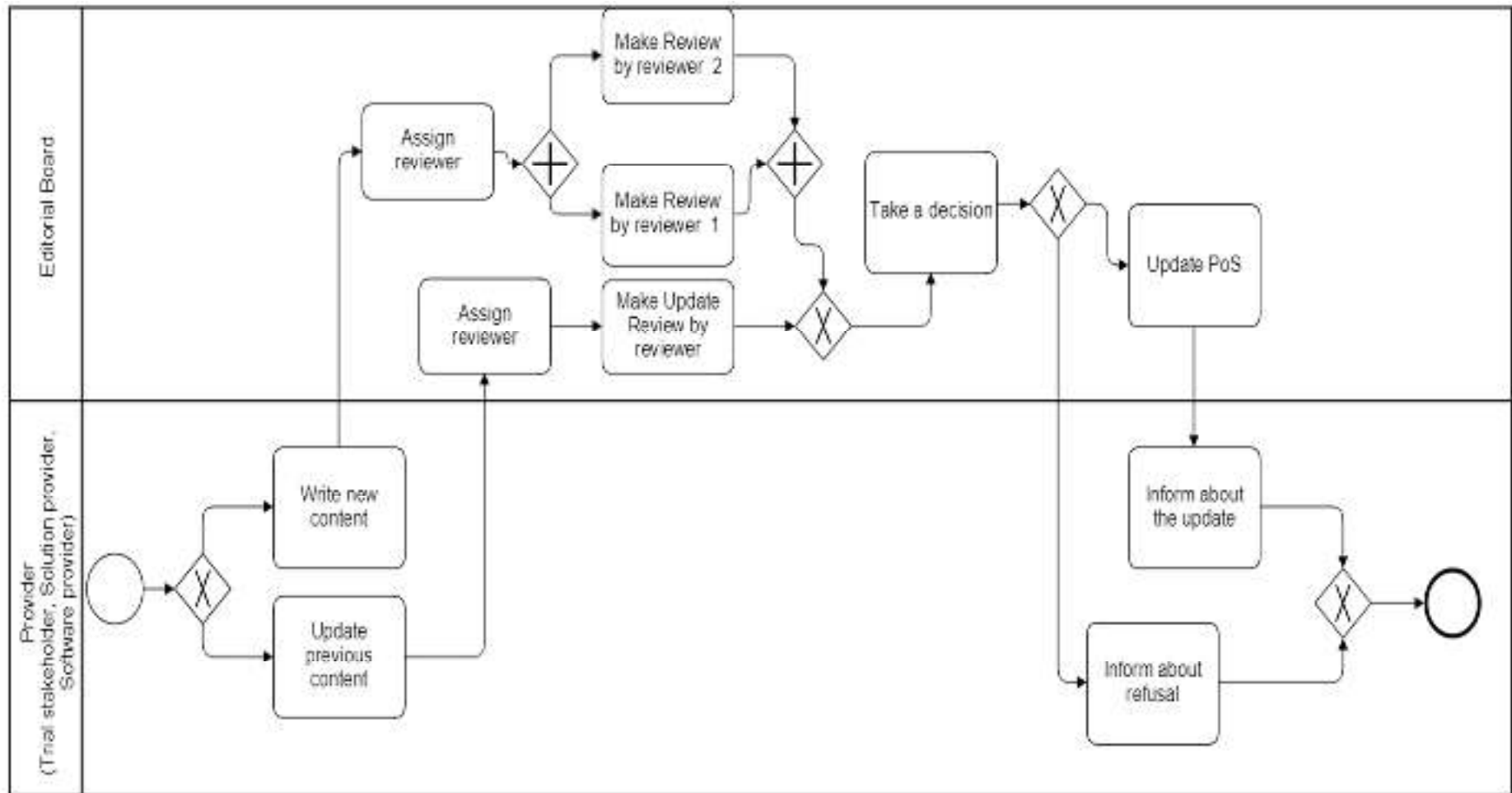


Figure 4.5: Content validation process

4.4.2 Solution integration validation process

Figure 4.6 presents the process that aims to ensure that a solution described in the PoS DB has been integrated with the Test-bed. This process will have to be used only for software solutions that need to exchange data with other software solutions or with the Test-bed.

A solution provider has to describe the connexion to the Test-bed and will provide a test case to validate the connectivity. Then, Test-bed users try to connect the software using the test cases (two test cases are needed when a connexion is described) and update the PoS DB with the result of the connection tests.

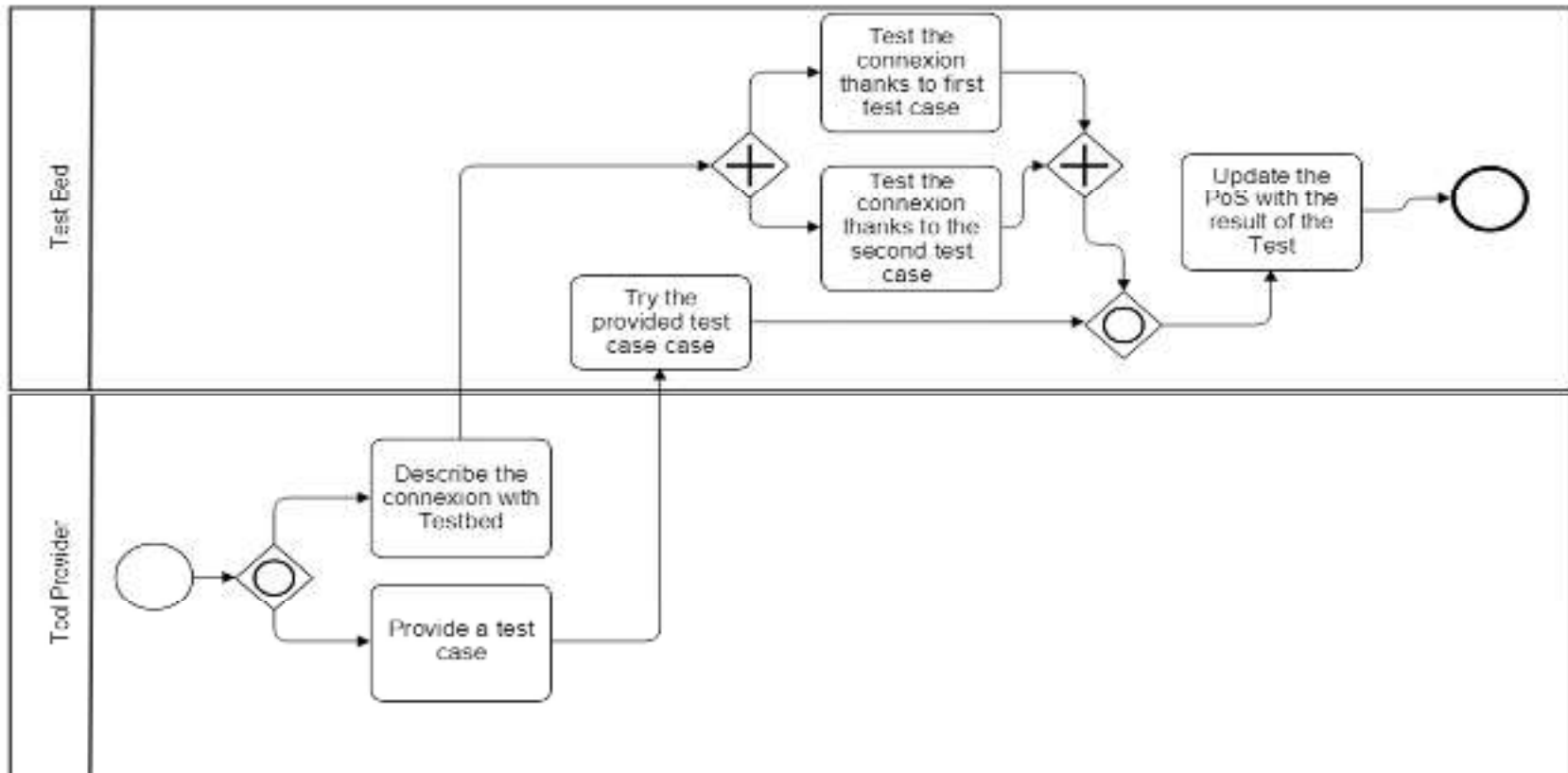


Figure 4.6: Solution integration validation process

4.4.3 Support of guidance methodology

The PoS DB will support processes and methodologies that are being defined by the guidance methodology team and that will be described in the following deliverables that are expected in the next stage of the project:

- D922.21: Trial guidance methodology and guidance tool specifications (version 1 due March 2018).
- D922.41: Trial guidance methodology and guidance tool specifications (version 2 due February 2019).
- D922.42: Handbook for systematic designing of Trials (due October 2019).

These processes are related to:

- Definition of the Trials.
- Preparation of the call for participation in a Trial.
- Submission of the application for participation in a Trial by solution providers.
- Pre-selection and final selection of the solutions.
- Definition the research questions, key performance indicators, and scenarios of a Trial.
- Execution of the Trials.
- Assessment of the results and compilation of the “lessons learned”.

The reader can refer to the above-mentioned documents for additional details once they are released.

4.5 Data model of the Portfolio of Solutions

In section 1, an overview of the relations between elements of the PoS database has been presented. These relations are summarized in Figure 1.3

Based on this overview and the requirements that are presented in section 4.2, it can be concluded that:

- Solutions include zero or more tools and at least one method.
- Solutions address one or more CM functions and/or CM gaps.
- Each Trial has one or more needs, and thus is divided in user stories.
- Each need (i.e.: User Story), relates to one or more CM functions and/or CM Gaps.

Regarding the relations and the expressed needs in previous sections, PoS DB has to ensure the link between the CM functions, solution, tool, method, Trial, etc. In order to achieve this goal, the use of meta-information is the selected solution. At least, the needed information is:

Dedicated data types for Trials, CM solutions, CM tools, CM functions, CM gaps, solution capabilities and Trial needs and requirements:

- Trials
 - Trial needs/requirements.
 - List of related user stories:
 - List of related CM functions.
 - List of DRIVER+ partners supporting a Trial list of related CM solutions:
 - Link to solution evaluation.
 - List or related CM methods:
 - Link to method evaluation.
- CM functions:
 - Description.
- CM Solutions:
 - List of solution capabilities.
 - List of related CM functions.
 - Solution provider.
 - List of related CM tools.
 - List of related CM methods.

- List of Trial results.
- CM Tools:
 - Tool provider.
 - TRL.
 - The available license.
 - Integrated or not in DRIVER+ Test-bed:
 - Link to the integration results.
- CM Gaps
 - List of related CM functions

All this data is the keystone of the PoS DB in order to achieve the objectives listed in section 3.

The data model proposed in Figure 4.7 will be improved through the actual development of the PoS taking advantage of the feedback gathered from its usage by the subsequent DRIVER+ Trials. It was done following the Merise formalism as defined by Avison (9). The objective of the data model definition is to normalize it so that each piece of the information is stored only once in the PoS DB. In this data model, several entities have been defined in order to store the information. Most notable features are:

1. Users of the PoS DB are separated into several categories, each of them having different roles and different access rights on the PoS DB. Depending on their rights, users can describe solutions, tools or user stories.
2. Solutions, tools, and Trials must also be part of the data model. A solution is composed of one or several tools, and the tools must be integrated into the Test-bed. This integration must be evaluated and registered.
3. Taxonomy must be provided in order to uniform the description and the understanding of solutions, tools and user stories. This taxonomy is the subject of the deliverable D934.10 “Taxonomy of CM functions for classification of solutions”.
4. Others entities are provided to store information about the country to which a user belongs, such as its resources, or the laws that apply.

It is important to notice that the current data model is likely to change significantly through the project, and is just a basis for the first development. In particular, the naming conventions have not been finalized yet and the lists of properties are not exhaustive and need to be completed during the project.

5. Information system design

This section is providing preliminary details of the actual information system design that will be done in WP933. It includes the information that is considered valid at the time of writing this deliverable and it will be refined in deliverable D933.11 “DRIVER+ online tools - Implementation specifications” (due M59).

The previous section describes the information that has to be included in the PoS DB as well as its lifecycle. This section describes the system design of the PoS DB from an informatics perspective.

A first high-level design decision to be taken is the selected approach to implement the PoS DB:

1. Initiate the development of a new web-based information management system.
2. Configure a Content Management System (CMS).

As the PoS DB has to be realized in a sustainable way, there is a need to select an existing CMS (option 2), having a strong community of users and to provide lots of configuration options.

Possible existing CMS with these features are WordPress, Joomla or Drupal. Drupal has been proposed as the CMS for the realization of the PoS DB taking into consideration that it is a CMS that can be easily extended using plug-ins (“modules”) and that has been extensively used with good results by members of the consortium. In the next section 5.1, a list of the selected plug-ins chosen to fulfill the expected functionalities of the PoS DB will be provided.

Initial tests and developments are currently performed using the Drupal 7, with the intention to upgrade to Drupal 8 later in the project.

5.1 Selected Drupal plug-in for the Portfolio of Solutions

Table 5.1 presents the plug-ins that are currently tested or scheduled for testing for their usability in the DRIVER+ context. This table is a result of initial development efforts of the PoS development team and therefore includes required features beyond those that are directly requested by the PoS users. Most notably, it includes the modules that simplify the development and administration of the Drupal websites.

Table 5.1: Drupal plug-in selected for the PoS DB

Required feature	Drupal module	Description	Development status
Advanced search	Alchemy or keyword research	This plug-in analyses the content of the website in order to extract the main keyword and provide a natural text research field (allows automated tagging of the content).	7.1 beta, no D8 version, not covered by the Drupal security advisory policy. (In short, this means that it cannot be used on the production server unless one decides to support the development of this module.)
	Search API	This module provides a framework for easily creating searches on any entity known to Drupal, using any kind of search engine.	7.x and 8.x stable versions of the Search API and of the Solr backend for the Search API module.
E-mail notifications	Mime mail	Provides basic support for sending MIME e-mails from a Drupal site (used by other modules. e.g., it can be used from rules for completely tailored solutions.	D7 stable, D8 alpha 2
	Notify	Enables notifications of new content and comments by e-mail.	D7 stable, no D8 version
Workflow support	Workbench	Workbench provides two functionalities: <ul style="list-style-type: none"> • A user interface for users who ONLY have to work with content. This decreases training and support time. • A customizable editorial workflow that integrates with the access control feature. 	D7 stable, D8 stable
	Workflow	This module allows you to create arbitrary workflows, and assign them to entities.	D7 stable, D8 rc1
Relate different entities	Entity reference	Provides a field type that can reference arbitrary entities	D7 stable, D8 core
	Corresponding entity references (CER)	CER keeps reference fields in sync. If you have two entities that refer to each other using entity reference (or some other kind of reference field), it saves you the trouble of double-editing your entities in order to have them point at each other.	D7 beta1, D8 alpha 2, not covered by the security policy.
Data import/export	Views data export	This module is designed to provide a way to export large amounts of data from views. It provides a display plugin that can render progressively in a batch.	D7 stable, D8 alpha 4

Required feature	Drupal module	Description	Development status
	User import framework	The User Import Framework (UIF) module provides simple, extensible user import via Comma-Separated Value (CSV) files.	D7 stable, no D8 port
	Data export import	The purpose of this module is to be able to export nodes, taxonomy terms and users to data files and then be able to import these data files into Drupal sites.	D7 stable, no D8 port
	Content synchronization	The content synchronization module provides a mechanism to export single content items, or all content items, from an environment, and move them to another, effortlessly.	No D7 version, D8 stable
Transfer features between development and operative site	Bundle copy	This module has export/import support for: node types, taxonomy user, field API fields, field groups	D7 stable, D8 core
	Features	The features module enables the capture and management of features in Drupal. A feature is a collection of Drupal entities which taken together satisfy a certain use-case.	D7 stable, D8 stable
Terminology / glossary support	Lexicon	The Lexicon module generates one or more Lexicon pages based on terms in taxonomies and optionally marks terms in the content and links them to the appropriate Lexicon page.	D7 stable but not maintained, no D8 version
	Taxonomy tooltip	This module allows you to show your term's description in the content in tooltip style. It means that it filters the terms (term words) in the content and shows their description in tooltip on mouse hover event.	D7 stable, merged with Glossify for D8
	Glossify	provides filters that scan and parse the content and replace terms in the text with links to their pages	D7 beta 1, D8 alpha 1
Custom views at the data	Views	Present the data in any way that is requested by the users	D7 stable, D8 core
	Field group	Provides compound fields that can be re-used as a group multiple times within any entity.	D7 stable, D8 beta 1 but the use discouraged (use paragraphs instead)
	Paragraphs	Similar functionality to field group but allegedly a “cleaner” implementation	D7 rc5, D8 stable
	View bulk operations	Allows users to perform various operations on multiple nodes at once.	D7 stable, D8 rc1

Required feature	Drupal module	Description	Development status
	Views field formatter	Provides a new field formatter: “View” for each field types and allows you to select a View and to use its result instead of the original field value.	D7 stable, D8 stable
	Views merge rows	Provides a way to combine rows with the same content in the specified fields.	D7 rc1, D8 stable
	Display suite	Display suite allows you to take full control over how your content is displayed using a drag and drop interface. Arrange your nodes, views, comments, user data etc. the way you want without having to work your way through dozens of template files.	D7 stable, D8 stable
Use of taxonomy	Taxonomy manager	This plug-in manages a hierarchical taxonomy and automatically underlines term of the taxonomy in any content of the website	
Single sign-on using external identity provider	Simple OAuth (OAuth 2.0)	Simple OAuth is an implementation of the OAuth 2.0 Authorization framework RFC.	D8 stable, no D7 version
	Services Oauth	Implementation of the Oauth 2.0 for D7	Deprecated, use https://www.drupal.org/project/services_v3.0 that is available in the stable version for D7 and in the beta4 for D8
	SimpleSAMLphp auth	This module integrates Drupal with SimpleSAMLphp, the most robust and complete implementation of SAML in PHP. It enables Drupal to communicate with SAML or Shibboleth identity providers (IdP) for authenticating users. The resulting Drupal site can effectively act as a SAML or Shibboleth service provider (SP).	D7 alpha, D8 rc3
Simple data input	Markdown filter	Provides support for the markdown notation	D7 stable, D8 stable
	CKeditor	WYSIWYG html editor	D7 stable, D8 core
	Quickedit (D8 Edit)	Provides in-place content editing.	D7 stable, D8 core. Does not work with any glossary filters in D7

5.2 Hierarchy of the Portfolio of Solutions website

Figure 5.1 describes the proposed hierarchy/structure of the PoS DB website. This hierarchy is divided into categories: CM solution, CM tools, user stories, CM functions, search, support, about and sign-In. Based on this description, search methods, as well as taxonomy, are used in order to facilitate the sharing of information.

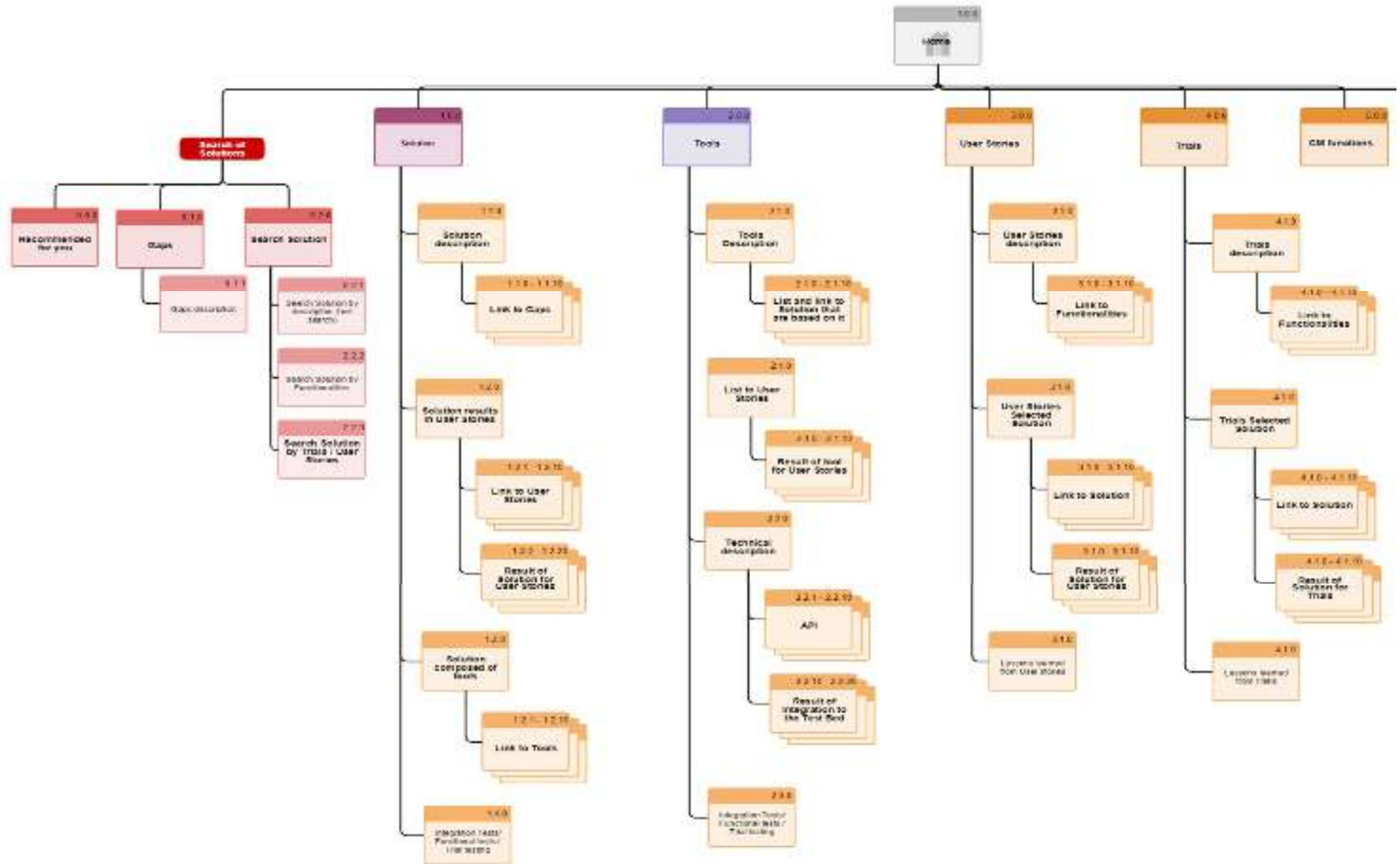


Figure 5.1: Hierarchy of the Portfolio of Solutions Database website

6. Conclusion

This deliverable aims at presenting the main expected functionalities of the PoS DB, based on the points of view of the different kind of users.

This design document is a result of extended discussions with PoS stakeholders within the project and requirement gathering that was organized during the first three months after the DRIVER+ project (re-) start. It specifies a list of the key functionalities that must or should be supported by the PoS DB according to input that was received from more than 31 team members with different backgrounds and verified by an even larger number of contributors at multiple teleconferences and live meetings.

In general, the requested functions aim to achieve the key objectives of the PoS DB:

- Help users share their understanding of crisis management, e.g. through the use of a shared vocabulary and predefined taxonomies.
- Allow the description of tools, solution and Test-bed elements, and the relation between them. Solution and tool providers will, through the PoS DB, be able to describe their element and to link it with CM functions and/or gaps. The tools and solutions will also be integrated to the Test-bed.
- Allow the design and execution of Trials, and the gathering of lessons learned from them. Thanks to the PoS DB, Trial owners will be able to design and execute Trials. The results the Trials will be stored in a database, which will enable them to be consulted.

The upcoming work related to this deliverable includes the realization of a first operational version of the PoS DB prototype (in WP933) and tutorials for using the site (WP932).

This first operational version will be used in Trial 1 (to be executed in June 2018) and its preparation work has already started (e.g.: Trial 1 call for applications). This is the reason why it is required that the high importance/high priority requirements listed in this deliverable will be implemented as soon as possible so that the DRIVER+ team can start using the site in their work.

Even more, the usage of the PoS DB during the preparation of Trial 1 will help to validate it and consequently detect any potential room for improvement that will be implemented for later Trials. This iterative process, in line with the Concept Development and Experimentation (CD&E) approach, is one of the main values of the DRIVER+ project.

The PoS DB will be further improved and enriched throughout the project as new requirements appear during the development or when the PoS DB will be tested and used during the Trials. User experiences from DRIVER+ partners as well as from external stakeholders will be taken into account.

References

1. **Bernus, Peter, Nemes, Laszlo and Williams, Theodore J.** *Architectures for enterprise integration*. s.l. : Springer Science & Business Media, 1996.
2. **Saadoun, Mélissa.** *Technologies de l'information et management*. s.l. : Hermes science publications, 2000.
3. *La modélisation des processus: typologie et proposition utilisant UML.* **Morley, Chantal.** 2002, Processus & Systèmes d'information.
4. **Morley, Chantal, et al., et al.** *Processus Métiers et SI: Evaluation, modélisation, mise en oeuvre*. s.l. : Dunod, 2005.
5. **Vernadat, F.** *Enterprise Modelling and Integration, principles and applications*. s.l. : Chapman & All, 1996.
6. *Systematic functional analysis methods for design retrieval and documentation.* **Zehtaban, L and Roller, D.** s.l. : Proceedings of World Academy of Science, Engineering and Technology, 2012.
7. *The five whys technique.* **Serrat, O.** s.l. : The five whys technique, 2017.
8. *MODEL, Business Process. Notation (BPMN) version 2.0.* **OMG.** 2011, OMG Specification, Object Management Group.
9. *MERISE: A European methodology for developing information systems.* **AVISON, D.** 1991, European Journal of Information Systems, pp. 183-192.

Annexes

Annex 1 – DRIVER+ Terminology

In order to have a common understanding within the DRIVER+ project and beyond and to ensure the use of a common language in all project deliverables and communications, a terminology is developed by making reference to main sources, such as ISO standards and UNISDR. This terminology is presented online as part of the Portfolio of Solutions and it will be continuously reviewed and updated⁶. The terminology is applied throughout the documents produced by DRIVER+. Each deliverable includes an annex as provided hereunder, which holds an extract from the comprehensive terminology containing the relevant DRIVER+ terms for this respective document.

Table A1: DRIVER+ Terminology

Terminology	Definition	Comment
Content Management System (CMS)	CMS is an application (more likely web-based), that provides capabilities for multiple users with different permission levels to manage (all or a section of) content, data or information of a website project, or internet/ intranet application. Managing content refers to creating, editing, archiving, publishing; collaborating on, reporting, distributing website content, data, and information.	
Crisis management function (CM function)		Definition is still “under construction” and can be found online in the near future.
Crisis management professionals	Persons with relevant knowledge or ability needed to effectively and timely respond to a crisis to in order to minimize damage to society.	
Gap	Gaps between the existing capabilities of responders and what was actually needed for effective and timely response.	
Glossary	Document or part of a document which contains a selected list of words from one or a few given texts which are deemed to require punctiform explanation but which explanations do not necessarily follow a uniform pattern.	
Practitioner	Individual person who ultimately benefits from the outcomes of the system (end-user).	
Solution		Definition is still “under construction” and can be found online in the near future.

⁶ Until the Portfolio of Solutions is operational, the terminology is presented in the DRIVER+ Project Handbook and access can be requested by third parties by contacting coordination@projectdriver.eu.

Terminology	Definition	Comment
Taxonomy		Definition is still “under construction” and can be found online in the near future.
Trial	An activity for systematically finding and testing valuable solutions for current and emerging needs in such a way that practitioners can do this in a pragmatic yet systematic way.	
User Story		Definition is still “under construction” and can be found online in the near future.

Annex 2 – User participation by activities

The following table lists the main contributors of this task during its activities apart from ARMINES, ATOS and AIT staff.

The list of participants of the KoM meeting dedicated to PoS has not been included due to their amount.

Table A2: List of participants

Users	Company	Type of user	Direct contributor	SP93 teleconference participant	Answer to the survey	SP92/93/94 meeting	Individual teleconference
Chiara Fiono	JRC	DRIVER+ project Test-bed stakeholders members	X	X		X	
Emil Wrzosek	SRC	Trial stakeholders	X	X	X	X	X
Todor Tagarev	CSDM	Standardization stakeholders	X	X		X	
Stine Bergersen	PRIO	Driver + community stakeholders	X	X	X		X
Alice Clemenceau	Valabre	Practitioners	X	X		X	
Andre de Rond	THG	Trial stakeholders	X	X			X
Laurent Dubost	TCS	Solution stakeholders		X	X	X	X
Wiktor Gawroński	SGSP	Trial stakeholders	X				
Camilo Palacio	ARC	Trial stakeholders	X	X		X	X
Adam Widera	WWU	Editorial Board	X	X		X	

Users	Company	Type of user	Direct contributor	SP93 teleconference participant	Answer to the survey	SP92/93/94 meeting	Individual teleconference
Thomas Obritzhauser	FRQ	Trial stakeholders	X				
Hector Naranjo	GMV	Editorial Board and Solution stakeholders		X			
Tim Stelkens-Kobsch	DLR	Editorial Board		X			
João Vale	EDISOFT	Solution stakeholders		X			
Chaim Rafalowski	MDA	Practitioners		X			
Angela Schmitt	DLR	Solution stakeholders		X			
Martha Bird	DRC	Practitioners		X			X
Martin Bjerregaard	DWR	Solution stakeholders		X	X		X
Ludwig Kastner	FRQ	Trial stakeholders		X			X
Rachel Beerman	ECORYS	Test-bed stakeholders				X	
Jean-Benoit Bonne	TCS	Solution providers				X	
Tarmo Kull	EASS	Test-bed stakeholders				X	
Marcin Smolarkiewicz	SGSP	Trial stakeholders				X	

Users	Company	Type of user	Direct contributor	SP93 teleconference participant	Answer to the survey	SP92/93/94 meeting	Individual teleconference
Grzegorz Taberski	ITTI	Trial stakeholders				X	
Joanna Tyminska	SRC PAS	Trial stakeholders		X		X	
Raul Perez Valencia	GMV	Solution providers		X		X	
Floris Jan van Brederode	XVR	Test-bed stakeholders				X	
Steven van Campen	XVR	Test-bed stakeholders				X	
Kees van Dongen	TNO	Test-bed stakeholders		X		X	
Job Verkaik	TNO	Test-bed stakeholders		X		X	
Erik Vullings	TNO	Test-bed stakeholders		X		X	
Gaby Gurczik	DLR	Solution Providers		x		x	
Georg Neubauer	DLR	Solution Providers		x		x	
Christian Niermann	DLR	Solution Providers		x		x	
Andrzej Adamczyk	ITTI	Solution Providers		x	x		

Annex 3 – Octopus diagrams to define PoS functionalities

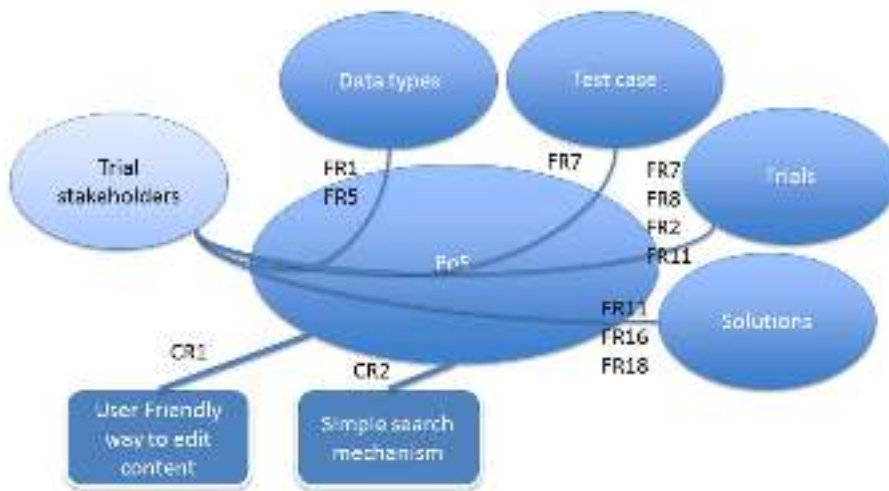


Figure A1: Octopus diagram for Trial stakeholders

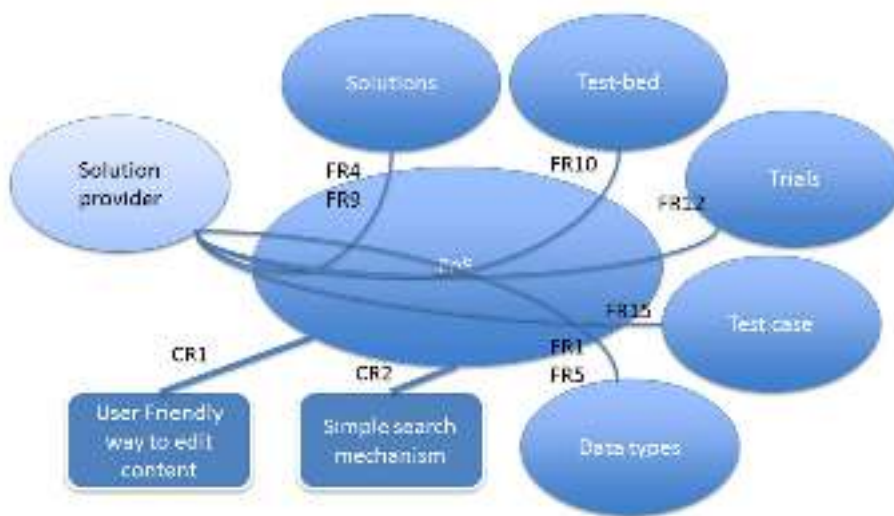


Figure A2: Octopus diagram for solution providers

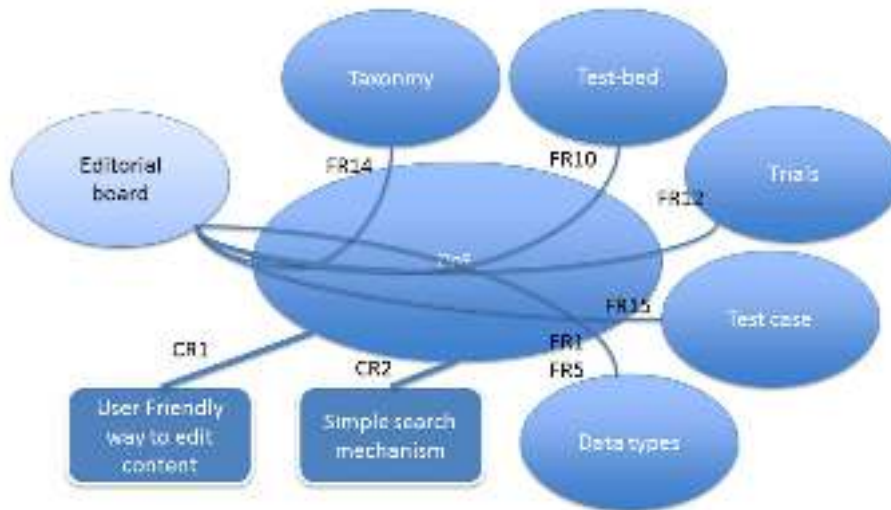


Figure A3: Octopus diagram for editorial board

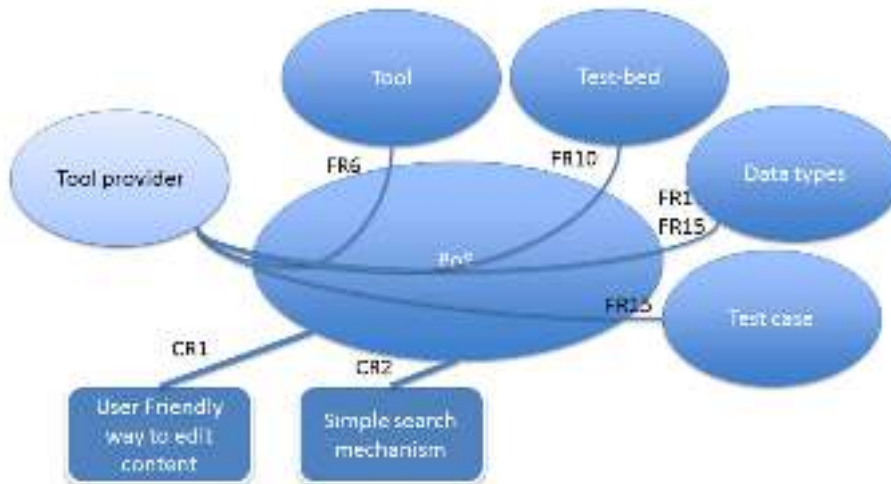


Figure A4: Octopus diagram for tool providers