ZDMP: Zero Defects Manufacturing Platform

ZERO DEFECTS Manufacturing Platform ZDMP

WP1: Management: Procedures, Metrics, Coordination, and Reporting

EU ID: D015: Technical Management – WP7 Report - Vs: 1.1A

ZDMP ID: D1.4.4

Deliverable Lead and Editor: Daniela Kirchberger, PROF

Contributing Partners: CET, UPV, IKER, ITI

Date: 2020-06

Dissemination: Public

Status: EU Approved

Abstract

The WP7 technical management overview report gives an in-depth view of the degree of completion of the software components developed in WP7. It shows the activities conducted so far, the planning, the main milestones achieved and the roadmap and general progress of those components. This report: M18

Grant Agreement: 825631

Document Status

Deliverable Lead	Daniela Kirchberger, PROF
Internal Reviewer 1	Moretón Álvaro, ROOT
Internal Reviewer 2	Cornel Ratiu, ALFA
Internal Reviewer 3	Francisco Javier Barrena, ITI
Internal Reviewer 4	Stuart Campbell, ICE
Type	Deliverable
Work Package	WP1: Management: Procedures, Metrics, Coordination, and Reporting
ID	D015: Technical Management – WP7 Report
Due Date	2020-06
Delivery Date	2020-06
Status	EU Approved

Project Partners:

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

Executive Summary

The main objective of “WP7: Process Quality Assurance” is to ensure the quality of the manufacturing processes along the value chain by developing zero defect manufacturing (ZDM) applications based on digital models of manufacturing assets and manufacturing processes. The deliverables of this work package and the WP1 Management work package are divided into software packages and document/reports. In terms of reporting:

- **Process/Status:** This report corresponds to D015 Technical Management: WP7 Report of WP1 Management: Procedures, Metrics, Coordination, and Reporting, and as identified in the DOA, focuses on the process/status of the work accomplished in WP7
- **Software:** All WP7 software deliverables of T7.1-T7.4 (type “OTHER”) are available in the ZDMP public repository with access details and install instructions as indicated in Section 1.4 and the component sections of this report. Each of these is accompanied by a very brief cover document. The deliverables are software components for: Digital Twin, Process Assurance Run-time, Prediction and Optimisation Run-time and Designer

The “WP7: Process Quality Assurance” consists of the following main parts: Resource flexibility, Process Quality Assurance, Material and Energy and Equipment Optimisation. The tasks of the WP are the following:

- T7.1 Preparation Stage: Start-up optimisation
- T7.2 Production Stage: Equipment Performance Optimisation
- T7.3 Production Stage: Material and Energy Efficiency
- T7.4 Process Quality Assurance

This document is structured in these four main sections (one per task). During the development period of the software components, according to the focus of each task, each component is assigned to one of the tasks. For example, prediction and optimisation designer from the point of view of models, up to this stage mostly is focused on creating and making accessible to developers a taxonomy of optimisation problems that are mostly relevant to start-up optimisation. However, after the development period, and as with other components in ZDMP, these components can be used (due to configurability) across different tasks of the work package. This explains prefixing multiple tasks numbers to the names of the predictions and optimisation designer and runtime components. In this period, this document reports on the following components:

- T7.1: T7.1/T7.2/T7.3/T7.4: Prediction and Optimisation Designer
- T7.2: T7.1/T7.2/T7.3: Prediction and Optimisation Run-time
- T7.3: T7.3: Digital Twin
- T7.4: T7.4: Process Assurance Run-time

Whilst from the reporting perspective each one describes planned activities, the progress, and the next activities for the period; from the software perspective an online public documentation is provided to describe the software functionalities and provides pointers to the location of the software developed so far.

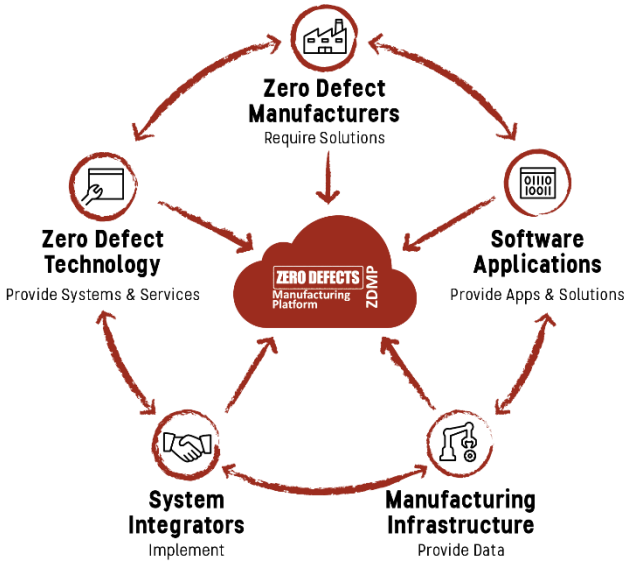
Table of Contents

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 16.2M€. 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, 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, allowing to achieve 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 optimize the overall processes regarding time constraints and product quality, the achieving the zero defects.

0.2 Deliverable Purpose and Scope

The purpose of "D015 Technical Management – WP7 Report" is to document the software produced in WP7 and to show the activities carried out so far, and plans, for the WP7 components, and the features available for each one of them.

This document is for internal reporting about scope of software components, mapping to architecture, planned activities, progress and limitations found. However, the report includes links to public repositories which complement and adds information about guidelines to set up a component, access to the service, installation notes, configurations details, requirements, etc.

Specifically, the DOA states the following regarding this Deliverable:

T1.4	**Technical Management: Leadership and Technical Reporting**			**ITI**	**M4-36**
D015 D016 D017	Technical Management: WP7 Report	R	PU	18, (24), 30, (36), (42) 48	RDI3, 5, 8
On the reporting side this task will deliver 6 monthly overview and per WP reports detailing the activity of T1.4 and T1.5 and WP5-8. Thus, these reports will be the conduit for the necessary paperwork reports related to the WP5-8 software deliverables since within these WPs the only deliverables are software prototypes accompanied by a cover page pointing to T1.4.x.x series of deliverables. Whilst six-monthly only the reports at the formal project periods will be presented to the Funding Authority and the bracketed reports are internal only. Such reports will include installation notes, initial guidance, and configuration details. Note the days to perform this module reporting will come from the RDI tasks themselves.					

0.3 Target Audience

The WP7 report deliverable aims primarily to the ZDMP managers in charge of producing the software components and zApps of the project, and the links to the public repositories are interesting for any party interested in the software produced.

0.4 Deliverable Context

Its relationship to other documents is as follows:

Primary Preceding documents:

- **D021 – Methodology Document:** It gives the foundations of collaborative software development and points to the different tools that are being used in the project
- **D048 – Requirements Document and Update:** It contains the end user requirements and those are mapped to the different ZDMP components
- **D051 – Global Architecture Specification and Update:** It depicts the first version of the architecture
- **D053 – Functional Specification and Update:** It contains the functional specification of ZDMP components and zApps
- **D055 – Technical Specification and Update:** It describes the different APIs of the components

Primary Dependant documents:

- **D006 – Technical management: Overview Report:** Describes the software development approach and provides integration guidelines among components

0.5 Document Structure

This deliverable is broken down into the following sections:

- **Section 1: Context:** An introduction to this deliverable including a general overview of WP7 components and their relationship with the overall architecture
- **Section 2: T7.1 Preparation Stage: Start-up optimisation:** Scope, link to public repository, features, progress, and planned activities for this component
- **Section 3: T7.2 Production Stage: Equipment Performance Optimisation:** Scope, link to public repository, features, progress, and planned activities for this component
- **Section 4: T7.3 Production Stage: Material and Energy Efficiency:** Scope, link to public repository, features, progress, and planned activities for this component
- **Section 5: T7.4 Process Quality Assurance:** Scope, link to public repository, features, progress, and planned activities for this component
- **Section 6: Risks:** Primary risks associated with the Work package
- **Section 7: Conclusions:** Main outcomes recap and lessons learned

- **Annexes**:
  - **Annex A**: Document History
  - **Annex B**: References

0.6 Document Status

This document is listed in the Description of Action as “public” since it provides information which may be relevant to external parties wanting to test or use the software produced within WP7.

0.7 Document Dependencies

This document is part of an iteration of living deliverables. This is the first version produced for M18. The following versions to be submitted for approval will be delivered in M30 and M48. Internal consortium updates of this report will be produced in M24, M36 and M42. Each subsequent version will provide updated information mainly, but not only, in the component overview. The additions will be clearly marked in further iterations of the document.

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

- Online documentation for each component is available at https://software.zdmp.eu/docs/components

0.10 Reading Notes

- None.

0.11 Document Updates

- This is the first version of this deliverable

1 Context

ZDMP's main aim is to provide 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 manufacturing. The ZDMP approach is modular by design and it is formed by a set of components which provides added-value features.

The features provided by WP7 components represent the elements dealing with digital models to automate and streamline the configuration of the equipment during the preparation stage (T7.1). During production, the developed models will provide a deeper prognosis of the real situation, which will be compared in real time with the data coming from the sensors and adjust to avoid errors (T7.2) whilst also considering energy and material use (T7.3). Finally, this work-package will introduce novel applications that use these digital models to guide process quality management towards the best possible result (T7.4).

1.1 Positioning in the Overall ZDMP Architecture

Figure 1 shows the global architecture. Each component fits within one tier:

- **Developer Tier (Design-time):** These components aide in the production of containerised applications for zero defect manufacturing - zApps
- **Enterprise Tier (Use-time):** These components assist the Run-time
- **Platform Tier (Run-time):** This is where zApps are installed. This component consists of Run-time services to provide a base level functionality for ZDMP
- **Edge Tier (Run-time):** This is composed of the Distributed Computing component which allows certain zApps to be run at various locations of the system for performance gains

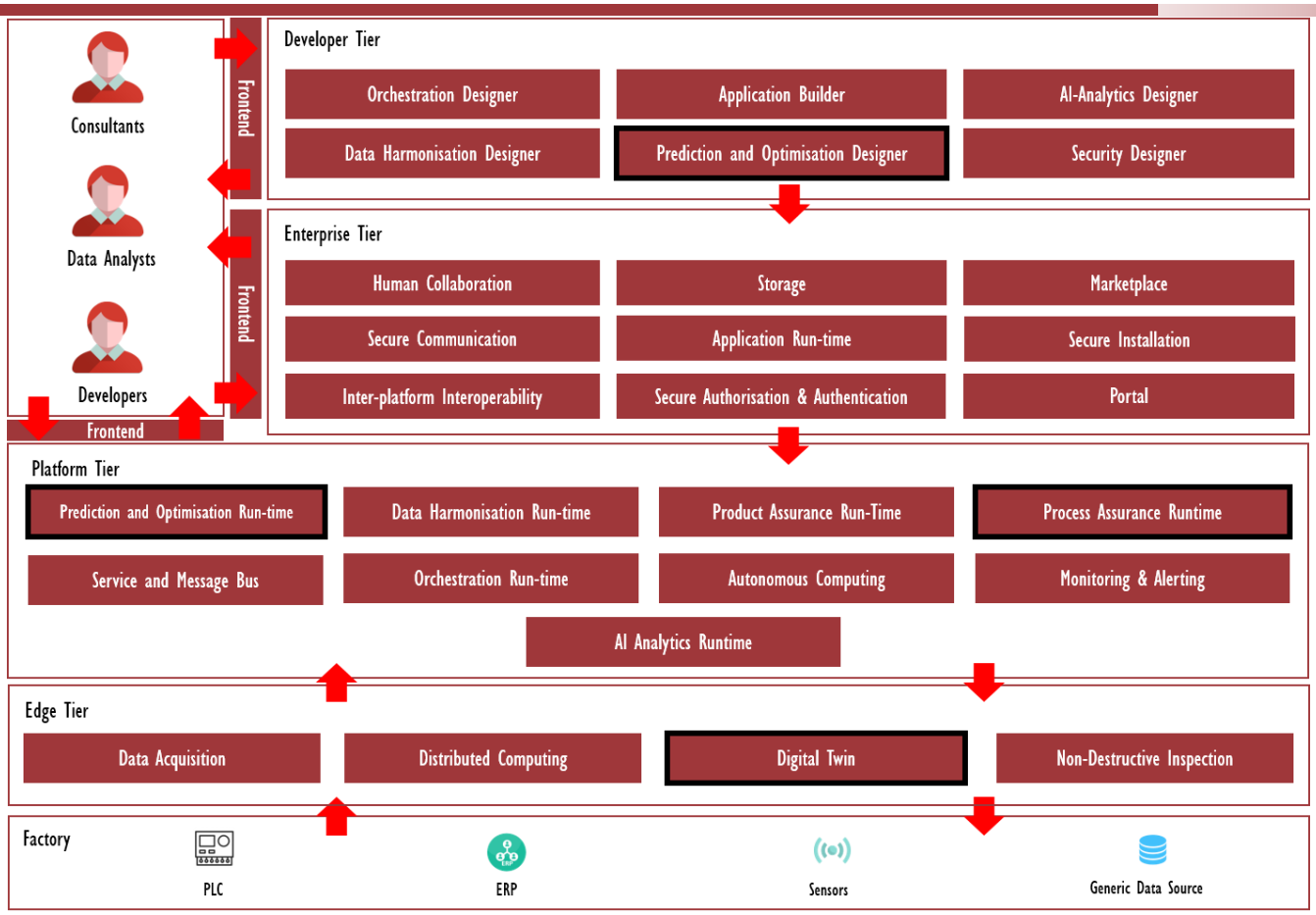


Figure 1: ZDMP Global Architecture; Components of WP7 are framed black

Figure 2 shows the architectural WP7 components covered by this report. In addition, it gives information about the different general qualities of each one of them.

Components	End-User Application in its own right	Callable Service	Intrinsic in the ZDMP Ecosystem	Standalone from the ZDMP Platform	Available in the Marketplace	Largely Created in Project
Developer Tier						
T7.1/T7.2/T7.3/T7.4: Prediction and Optimisation Designer	✔			✔		✔
Platform Tier						
T7.1/T7.2/T7.3: Prediction and Optimisation Run-time		✔	✔		✔	✔
T7.4: Process Assurance Run-time		✔	✔		✔	✔
Edge Tier						
T7.3: Digital Twin		✔	✔		✔	✔

Figure 2: WP7 ZDMP Components Classification Table

All components of WP7 share the same qualities:

- **Callable Service:** Typically, applications may be downloaded and put in a user environment; however, services may then be called as they may run solely in someone else's environment – eg in another factory. However, care must be taken

since the boundary is blurry – for example, an application is created/downloaded and run as a service for another party then it could fall into both or either camp

- **Intrinsic in the ZDMP Ecosystem:** This implies that the project builds/buys this service/component and it would (always) form part of the platform
- **Available in the Marketplace** Self-explanatory
- **Largely Created in Project:** Self-explanatory

1.2 Purpose of the WP7 Components

It is important to define the purpose of the ZDMP components to set the scene where these components are being developed:

- **T7.1/T7.2/T7.3/T7.4 Prediction and Optimisation Designer (PO Designer):** Allows developers to discover, use in their ZDM applications, and share with other developers the resources to implement new zero defects solutions based on the prediction and optimisation Run-time. Developers can also share algorithms and machine learning models to be adapted or reused and datasets to test and validate new solutions
- **T7.1/T7.2/T7.3 Prediction and Optimisation Run-time (PO Run-time):** Provides computational services such as prediction, simulation, and optimisation on the one hand to the platform as a whole and on the other hand to T7.4 Process assurance run-time and to T7.3 Digital Twin
- **T7.3 Digital Twin:** Defines the structure and contextualization of the elements of any kind of industry: assets, products, and processes and its mathematical behaviour inside a Simulation subcomponent. The Digital Twin component base functionalities (nodes, attributes, signals, static values, etc) allows modelling both processes of WP7 and products of WP8. In the WP7 context, the Digital Twin component is focused on processes modelling and establishes a layer for bridging with product model on WP8
- **T7.4 Process Assurance Run-time (PA Run-time):** Contains operations that independently or in synergy with Digital Twin provides process assurance services

1.3 Metrics

The metrics from the DOA are as follows:

Result	The results of this WP will be the delivery of solutions, heavily based in machine learning, to support self-configuration and start-up optimisation (T7.1), solutions to support self-Optimisation and self-healing during production (T7.2), and considering energy and materials consumption (T7.3). The final deliverable (T7.4) will deliver management applications that will use and orchestrate these solutions to control process quality.	
Metric	• Number of models consistently covering the use cases • Number of applications for zero-defects manufacturing at process level • Process quality improvement • >= 10 models for industrial process start-up quality and error control • >= 10 models for industrial equipment performance control	• >= 10 models for industrial process energy efficiency control • >= 10 models for industrial process materials efficiency control • >= 5 applications for manufacturing process quality management • >= +10% productivity increase through process start-up optimisation • >= +10% energy/material resource efficiency increase via process optimisation

The status is as per the table below. The focus of the first months was the setup of the software architecture and the framework requirements for the development of the above-mentioned models and improvements. Therefore, the numbers of the metrics are low at this current state.

Metric	Related Task(s)	Target	M18	M30	M48
Number of models covering the usecases	All	N/A	2		
Number of applications for zero-defects manufacturing at process level	All	N/A	1		
Process quality improvement (in %)	T7.4	N/A	0		
Models for industrial process start-up	T7.1	>= 10	1		
Models for industrial equipment performance control	T7.2	>= 10	3		
Models for industrial process energy efficiency control	T7.3	>= 10	0		
Models for industrial process materials efficiency control	T7.3	>= 10	0		
Applications for manufacturing process quality management	T7.4	>= 5	0		
Productivity increase through process start-up optimisation (in %)	T7.1	>= +10	0		
Energy/material resource efficiency increase via process optimisation (in %)	T7.3	>= +10	0		

1.4 Components public documentation

Technical documentation of the components is publicly available in the following URL:

https://software.zdmp.eu

The public documentation includes for each one of the components the following information:

- **Source code:** Location
- **Latest release:** Link to download and numbering version
- **Open API Spec:** Available Open API Specifications for the component
- **General Description:** Description of the use of the component in the project general scope
- **Features:** List of features provided by the component
- **Requirements:** Technical requirements of the component (software and hardware)
- **Installation:** Description of the steps to install the component
- **How to Use:** Instructions of how to set up and use the component

1.5 Risks

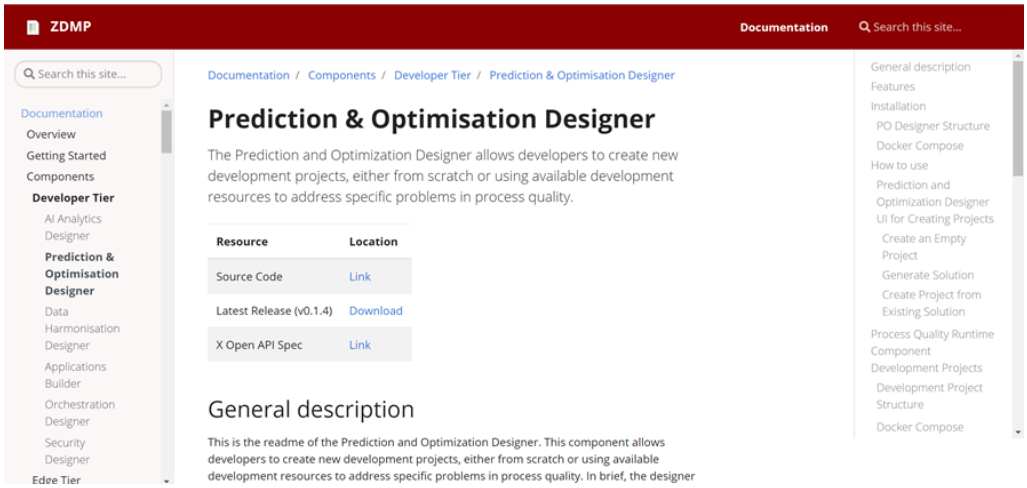
For the status of risks related to the work package see Section 6. For project-orientated technology risks in general, please refer to the following document: D006 – Technical management: Overview Report.

2 T7.1 Preparation Stage: Start-up optimisation

2.1 Public documentation

Public documentation is available as indicated in Section 1.4. Regarding this component the description is available at:

https://software.zdmp.eu/docs/components/developer-tier/prediction-and-optimisation-designer/



CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/developer-tier/t7.1-t7.2-t7.3-t7.4--prediction-and-optimization-designer
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
System Requirements:	Available	N/A
Installation:	Available	N/A
How to Use:	Available	N/A

2.2 M18 Report

2.2.1 Architecture Implementation

In this period the focus is on the design time and more specifically, the development of optimisation and simulation techniques for start-up optimisation in diverse types of production systems. These modules are included as part of the Preparation Stage Template Repository. Additionally, in this period there are prototypes available for the design wizard that will guide developers in the creation of new instances of the runtime. The wizard will also be used to build runtime instances for other stages (eg production), but to enable traceability and management they have been assigned to T7.1 Preparation Stage. Figure 3 below shows the status of the architecture implementation:

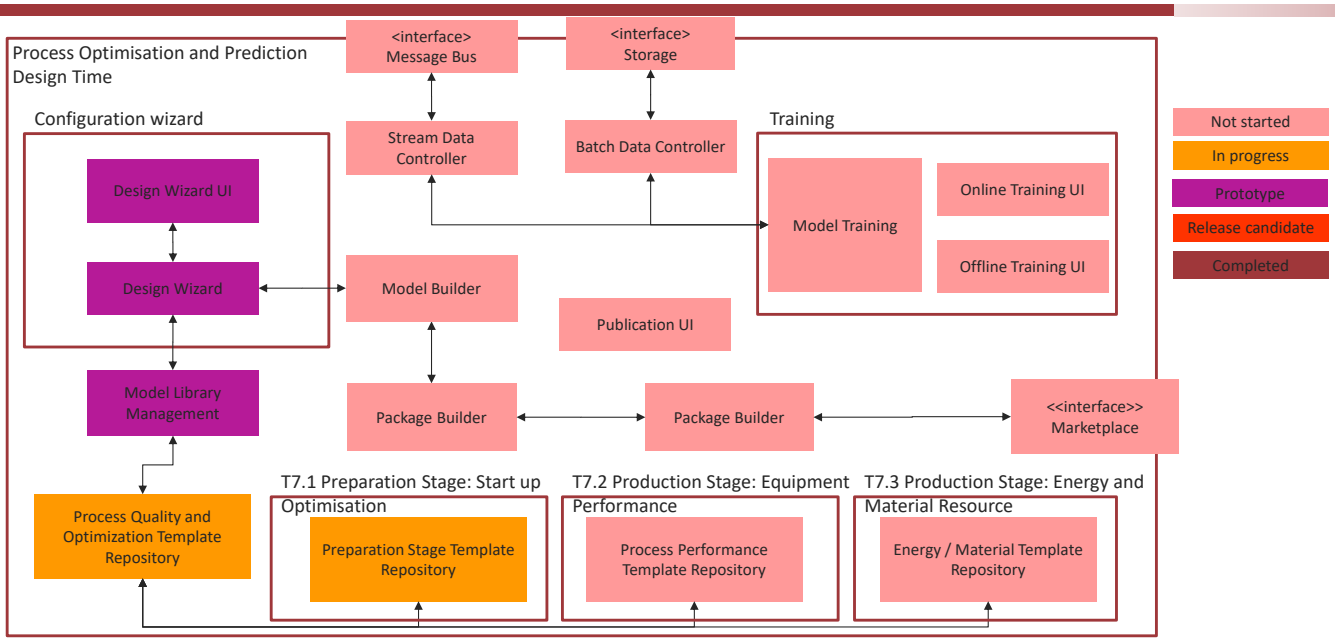


Figure 3: Process Optimisation and Prediction Design Time

2.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Classification of Process Quality problems for the designer.** To define a taxonomy of process quality problems that can be used to help developers find development resources useful to solve specific process quality problems. The classification is used in the different design user interfaces to search development resources by type of problem or problem domain (for instance Bayesian Networks and Knowledge-based Systems are solutions linked to a process indicator out-of-range problem in the process quality detection domain). The taxonomy is described in detail in the online documentation (see Section 1.4)
- **[Completed] Provide integrated PO Designer UI**. Web application tool used for generating development projects to solve zero defects manufacturing problems. The PO Designer User Interface is structured into three main functions:
  - Create projects from scratch so that developers get a clean solution to build models and incorporate datasets to solve process quality problem.
  - Develop solutions providing their algorithms based on existing datasets that will be used for validation
  - Create a project from an existing solution (models and dataset layers)
- **[Completed] UI for filtering Algorithms by additional criteria (performance, interpretability)**. Web components to search existing solutions according to criteria such as performance and interpretability of the algorithms developed
- **[Completed] Classification for Algorithms that can solve Process Quality Problems**. This task covers the definition of a zero defects manufacturing taxonomy covering domains, problems, and solutions. The taxonomy considers three stages at the manufacturing process in which quality problems could be addressed: Pre-execution stage, execution stage, and post-execution stage
- **[Completed] UI for Searching and filtering quality solutions by objectives**. Web components to search and filter by the category of business objectives categorised in

four main categories: Financial objectives, process objectives, customer objectives, and environmental objectives

- **[Completed] Backend for retrieving and filtering the list of quality solutions by objectives**. The PO Designer backend provides functionality to store and retrieve the taxonomy of business objectives that are used by the PO Designer UI to search and filter existing ZDM solutions
- **[Completed] Classification for Process Quality problems for Designer**. A taxonomy classifying potential problems associated with manufacturing processes has been defined through an ontology using WebProtege. This categorisation of process quality problems such as Process Indicator Deviation or Process Quality degradation is useful to index ZDM techniques that users can use to solve them
- **[Ongoing] PO Designer UI integrated API**. The PO designer provides a backend supporting all the functionality offered by the UI. This includes querying and filtering process quality domains, problems, and solutions, navigating information registered in existing solutions and datasets to inform the developer about a range of information such as complexity, business objectives and interpretability for instance
- **[Ongoing] UI for Algorithm registration**. A web application to register new algorithms
- **[Ongoing] Backend for filtering Algorithms by additional criteria (performance, interpretability)**. The Prediction and Optimisation designer backend provides functionality to retrieve and filter solutions from the metadata tagging them
- **[Ongoing] Populate Classification of Quality Problems with Pilots cases**. This task covers the identification of process quality problems in the pilot cases and the subsequent mapping of those problems into the process quality problem taxonomy. In that way, techniques associated with solving process quality problems will be available for their application
- **[Ongoing] Provide Optimisation template and Prediction template for Generator**. To provide a source code template to generate prediction and Optimisation Run-time components
- **[Not started] Generator for Optimisation and prediction components**. To implement a generator to create an optimisation/prediction component. The generator will receive the input from the UI Designer to build a configured and ready-to-run Prediction and Optimisation Run-time Component

2.2.3 Progress

The development team was managed based on agile methodologies, conducting sprints of 2 weeks of duration. The following table shows the progress made by the developers per sprint. When applicable, sprint issues are linked to features specified in the functional specifications (D053 – Functional Specification and Update). Note that the “PEnnn” (or similar) refers to the unique identifier assigned to each feature in the functional specification and if there is no PEnnn it implies that the issue is related to a new feature to be included in a future revision of the functional specification or to a non-functional requirement. The documentation is only (generally) up to the last full sprint in May; this point was chosen as the last feasible documentation point to be able to produce and review these documents. The next report will follow on after this point.

Sprint 1 (Jan 27, 2020 – Feb 10, 2020)
• **Taxonomy for problem classification.** Define a classification of problems to guide the wizard. Related Subtasks: PE001 Model Project

- **Problem data model.** Define a classification of problems to guide the wizard. Related Subtasks: PE001 Model Project

Sprint 2 (Feb 10, 2020 – Feb 23, 2020)

- **Taxonomy of objectives.** First version of the taxonomy of objectives. PE001 Model Project
- **Solution taxonomy.** Define a classification of techniques to browse the manager libraries along with the problem taxonomy and the objective function. Related Subtasks: PE003 Set Time Performance
- **Objective Selection backend.** Develop API backend service to get Objectives**.** Related Subtasks: PE002 Select Optimisation Objective
- **Object Selection UI.** Develop API backend service to get Objective. Related Subtasks: PE002 Select Optimisation Objective
- **Problem data model.** Populate taxonomy with WP problems and objectives. Related Subtasks: PE002 Select Optimisation Objective
- **Select Objective API call-back.** Develop API backend service call-back function to get Objectives. Related Subtasks: PE002 Select Optimisation Objective
- **Object Selection UI.** Implement the data model for problems. Related Subtasks: PE002 Select Optimisation Objective

Sprint 3 (Feb 24, 2020 – Mar 08, 2020)

- **Swagger file definition.** Create the swagger file with the definition of the backend API.
- **Fix frontend file structure.** Merge PO_Designer into frontend
- **Fix readme file.** Include short installation and usage guides

Sprint 4 (Mar 09, 2020 – Mar 22, 2020)

- **Usage page**. Refine the usage documentation describing the current version
- **Dataset and layer contribution pages.** Documentation pages to describe how to add a new dataset and a new layer
- **Resource UI component.** Add a new component to present the resources that match the search criteria. Since there are going to be in general few solutions (at least for now) can use a "card" component to present some basic information about the component and two controls**:**
  - Select to add to development project
  - Check details to see detailed information
- **New button to apply filters.** New button to apply filters. It updates the chipsets, resets the selection of the filter components, and closes the overlay panel
- **Add search button.** Change icon of button to display overlay panel with filters to a down arrow and add a new button to apply filters. This button will call the backend to filter the results and update the list of resources
- **Fix issue with multiple chips in search bar.** Fixed issue that did not allow to add several chips in search bar when different filter criteria are applied simultaneously
- **OWL ORM and semantic reasoning.** Refactor services to load the .owl ontology into a Python Object and reason with it in the backend service. The [Owl ready 2] (https://pypi.org/project/Owlready2/) module allows to load an owl file and reason with it in Python using reasonres (Hermit and Pellet are included)
- **Export webprotégé ontology.** Export current version of the ontology as a .owl file and import it to the project folder data
- **UI wizard backend integration.** Integration with the backend services. Related. Subtasks: PE001, PE002, PE003, PE004, PE005
- **UI wizard mock-up.** Wizard-like UI mock-up to create projects. Related Subtasks: PE001, PE002, PE003, PE004, PE005
- **Backend docker file.** Docker file for the backend Related Subtasks: PE001, PE002, PE003, PE004, PE005
- **Resource data model**. In this Sprint a simple json can be used to model each resource. Implement the data model for solutions. Related Subtasks: PE006 Set Algorithm
- **Algorithm Selection UI.** Develop API backend service (swagger) Related Subtasks: PE005 Get Algorithms

• **Performance Selection UI.** Develop UI components to select techniques parameter settings**.** Related Subtasks: PE004 Set Accuracy
Sprint 5 (Mar 23, 2020 – Apr 05, 2020)
• **Documentation.** Fixes to online documentation material. Additional documentation materials and demo environment for plenary • **Plenary Technical Meeting**
Sprint 6 (Apr 06, 2020 – Apr 19, 2020)
• **Workforce scheduling heuristic.** Develop heuristic that solves the Workforce scheduling problem (scheduling of project activities with workforce and material resources) • **Workforce scheduling heuristic adaptation.** Develop wrapper function to conform to layer specifications so that it can be used in runtime • **Workforce scheduling heuristic containerisation.** Development and tests of Dockerised version of the layer • **Workforce scheduling datasets.** Create datasets to develop and test heuristics for Workforce scheduling. A small dataset to verify that the integrated algorithm works. A large dataset to optimise performance in a large-scale deployment scenario and a dataset obtained from a MS Project file provided by the construction pilot owner • **Workforce scheduling heuristic integration.** Deployment of the layers in runtime and deployment of datasets in mock file storage service to perform integration tests. • **Workforce scheduling readme file.** Write down Readme of the Workforce scheduling problem model and a description of the heuristic
Sprint 7 (Apr 20, 2020 – May 04, 2020)
• **Documentation**. Changes into documentation and alignment to CI specifications • **Jopshop simulator.** Development of a Python module that implements discrete events simulation models of the execution of production orders in a Job-shop • **Jopshop simulator adaptation.** Develop wrapper function to conform to layer specifications so that it can be used in runtime • **Jopshop simulator containerisation.** Development and tests of Dockerised version of the layer • **Flowshop genetic algorithm:** Development of a Python module that implements a genetic algorithm to solve a flow shop production order sequencing problem

2.2.4 Limitations

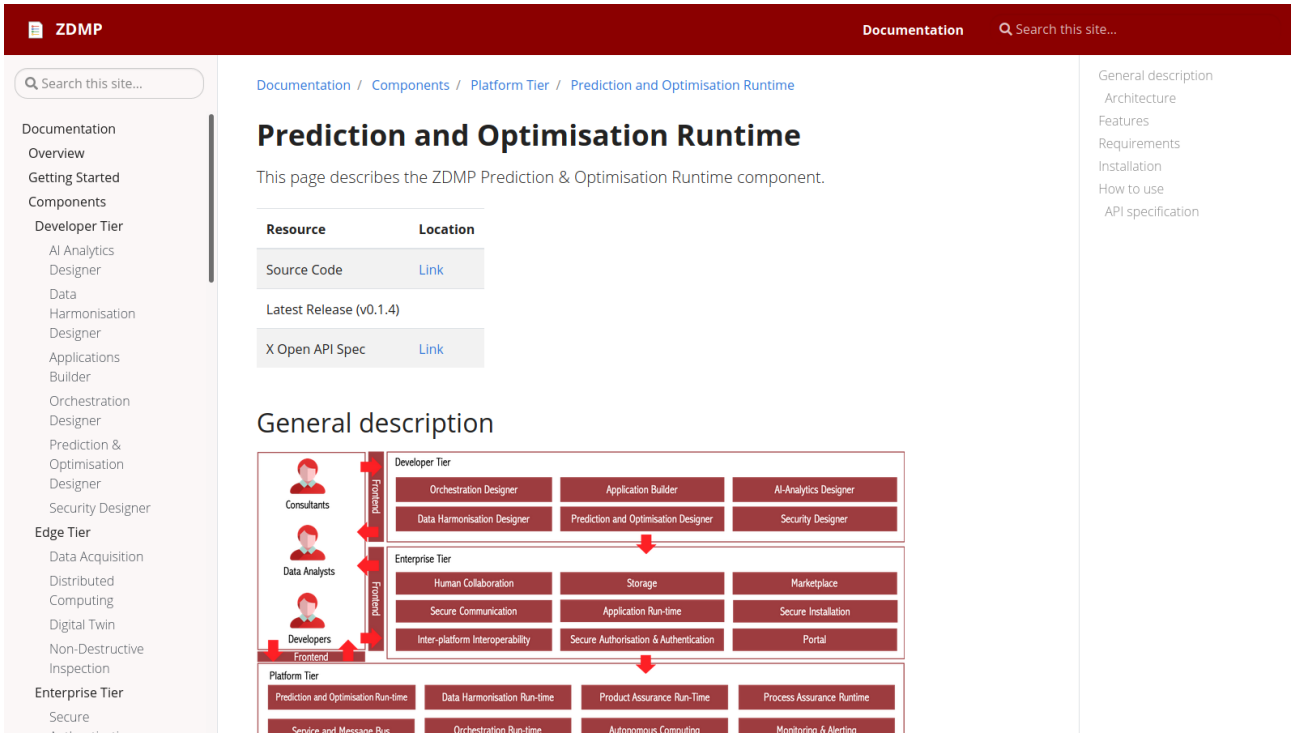
There were no issues found in this period which would limit the final scope of the activities expected.

3 T7.2 Production Stage: Equipment Performance Optimisation

3.1 Public documentation

Public documentation is available as indicated in Section 1.4. Regarding this component the description is available at:

https://software.zdmp.eu/docs/components/platform-tier/prediction-and-optimisation-runtime/



CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/platform-tier/t7.1-t7.2-t7.3-prediction-and-optimization-run-time
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	N/A
General Description:	Available	N/A
Features:	Available	N/A
System Requirements:	Available	N/A
Installation:	Available	N/A
How to Use:	Available	N/A

3.2 M18 Report

3.2.1 Architecture Implementation

The Prediction and Optimisation Run-time (PO Runtime) component consists of a uWSGI Sever that deploys a Python Flask application as its main service. In the current version, also a MongoDB database as well as a MQTT Broker (Eclipse Mosquitto) are deployed to persist configurations and serve test data. Later these services will be replaced by the ZDMP Storage component and the ZDMP Message-Bus respectively. The techniques for equipment performance optimisation are developed as independent Python packages that are later injected and used in the PO Run-time.

Figure 4 below shows the status of the architecture implementation:

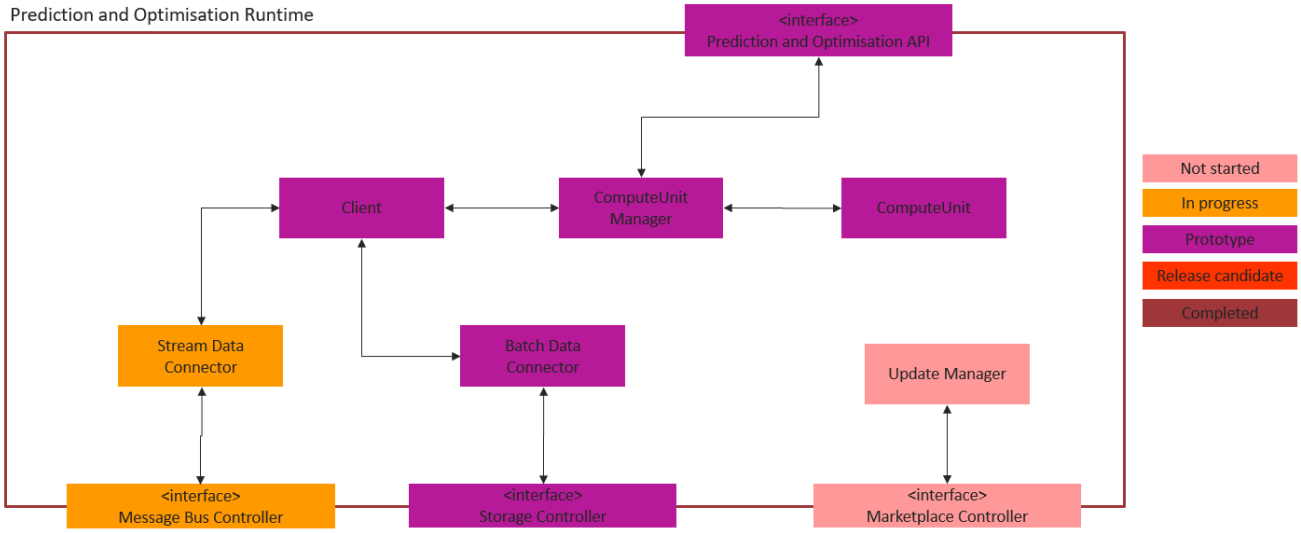


Figure 4: Prediction and Optimisation Run-time

3.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Simple optimiser:** Implement the optimisation sub-component which can start optimisation calculation and provide the optimisation manager with the results
- **[Completed] Define Backend Architecture:** Define a docker container for an uWSGI Server that uses a Flask App
- **[Completed] Define Frontend Architecture:** Define a docker container with a Nginx Server that serves a React UI
- **[Completed] Storage simulator:** Since the storage component of the platform is not available yet, implement the Storage module. In the first version it will only support CSV files
- **[Completed] Simple predictor:** Implement a simple prediction model, such that users of the component can connect to the component and by providing the necessary inputs request a prediction.
- **[Completed] Optimiser manager:** Implement the optimiser manager module which responds to the API calls and initializes optimisation operation by calling the optimisation subcomponent (which implements the actual mathematical optimisation)

- **[Completed] Prediction manager:** Implement the prediction manager module which responds to the API calls and initializes prediction operation by calling the prediction subcomponent (which implements the actual mathematical prediction operation)
- **[Ongoing] UI for predictor:** Implement the UI for the predictor which can receive inputs from a human user and send prediction requests to the prediction manager
- **[Ongoing] Define API**: Define the Open API Specification of the Python backend
- **[Not started] UI for a specific optimiser:** Implement the UI for a specific optimiser which can receive inputs from human user and send optimisation request to the optimisation manager

3.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 1 (Jan 27, 2020 – Feb 10, 2020)
• **Docker compose.** Define a docker compose file for the initial overall project structure • **Define Frontend Architecture.** Define a docker container with a Nginx Server that serves a React UI • **Define Backend Architecture.** Define a docker container for an uWSGI Server that uses a Flask App. Related Subtask: T7123A004 • **Backend Skeleton.** Define the modules and empty methods for the Connexion Python backend that implements the API. Related Subtask: T7123A004
Sprint 2 (Feb 10, 2020 – Feb 23, 2020)
• **Use onnx for model saving and loading.** Related Subtask: T7123A004 • **Backend as an installable python package.** Make the source code as an install-able package. • Prediction manager. Implement the prediction manager module • **Simple predictor.** Implement a simple prediction model. Related Subtask: T7123A004 • **Create a first UI that displays the results from a specific predictor.** Implement a first UI that: - displays the results returned by a specific
Sprint 3 (Feb 24, 2020 – Mar 8, 2020)
• **Write test for pre-processing model and module.** Test for the pre-processing model and prediction manager • **Load and use saved pre-processing model.** In the initialization step load the model based on configuration data. Related Subtask: T7123A005 • **Create and save pre-processor model.** Create pre-processing model such that it could be exported to onnx format and create the actual onnx file. Related Subtask: T7123A004 • **Add logging.** Create an application-wide logger and output info and debug log in prediction manager and predictor • **Make model info in prediction manager reconfigurable at the start and on the fly.** The model info (eg model filename) should be configurable and reconfigurable-on-the-fly. Related Subtask: T7123A005 • **Storage simulator.** Implement the Storage module. In the first version it only supports CSV files. Related Subtask: T7123A003
Sprint 4 (Mar 9, 2020 – Mar 22, 2020)
• **Analysis of the PO Run-time structure and the way to implement the models in it.** Study of the layer subsystem and how to use it • **Database simulator manual:** Short manual about how to use the database simulator, db. Client. Related Subtask: T7123A003 • **Short documentation about the layer subsystem and how to use it along with its docker container and docker compose.** Show the usage examples for running the containers given multiple docker compose files in the orchestration folder • **Adapt to optimiser layer:** Backend work with layered optimiser. Related Subtask: T7123A003 • **Adapt to added service:** Backend work with added storage service**.** Related Subtask: T7123A003

- **UI: Add additional views:** add table/graph to analyse/view selected data sources. Related Subtask: T7123A001
- **UI: Data source:** Integrate data source selection into the UI. Related Subtask: T7123A001
- **Analysis of different algorithms for data processing.** Study of the algorithms that best adjust to the data available for prediction and Optimisation.
- **Data Selection for quality and prediction analysis.** Study of the correlation between the different machine tool parameters to obtain the best ones to characterize and optimize the process. Related Subtask: T7123A003
- **Write test for prediction manager when address of resource is given.** Batch data connector should read from a previously created dummy database. Related Subtask: T7123A001
- **Definition of usage procedure for the zApps.** Discussion, with the final end-users, of the functionality for the zMachineAnalytics and the zMachineMonitor
- **Studying the contribution to the shared sub-components**

Sprint 5 (Mar 23, 2020 – Apr 5, 2020)

- **Refactor Predictor**: Change predictor sub-component to Layer system. Related Subtask: T7123A004
- **Debug Frontend:** The frontend doesn't respond when started in Docker. Possibly due to cross-origin request. Related Subtask: T7123A004
- **Plenary Technical Meeting**

Sprint 6 (Apr 6, 2020 – Apr 19, 2020)

- **Update API - Allow configuration of PO Run-time.** Merge Optimisation & prediction API. Add get/set parameters**.** Related Subtask: T7123A005
- **Move frontend to the Process Assurance Run-time component.** Move the frontend and Modify the dependent modules accordingly
- **Move ONNX predictor to "Layers".** Create a layer container and corresponding package which works with models exported as onnx. Related Subtask: T7123A004

Sprint 7 (Apr 20, 2020 – May 3, 2020)

- **Update API - Merge prediction and optimisation API.** Update the API based on abstract “compute” model. Related Subtask: T7123A005
- **Merge prediction- and optimisation-manager.** Based on new “compute” model create a higher level “compute manager”. Related Subtask: T7123A004
- **Workforce scheduling heuristic.** Heuristic that solves the workforce scheduling problem**.** Related Subtask: T7123A003
- **Workforce scheduling dataset.** Small dataset to develop and test heuristics for workforce scheduling**.** Related Subtask: T7123A003
- **Workforce scheduling readme file.** Readme of the workforce scheduling problem model and a description of the heuristic**.** Related Subtask: T7123A003
- **Stochastic Gradient Descent Optimiser:** Load data from CSV files. Related Subtask: T7123A003

3.2.4 Limitations

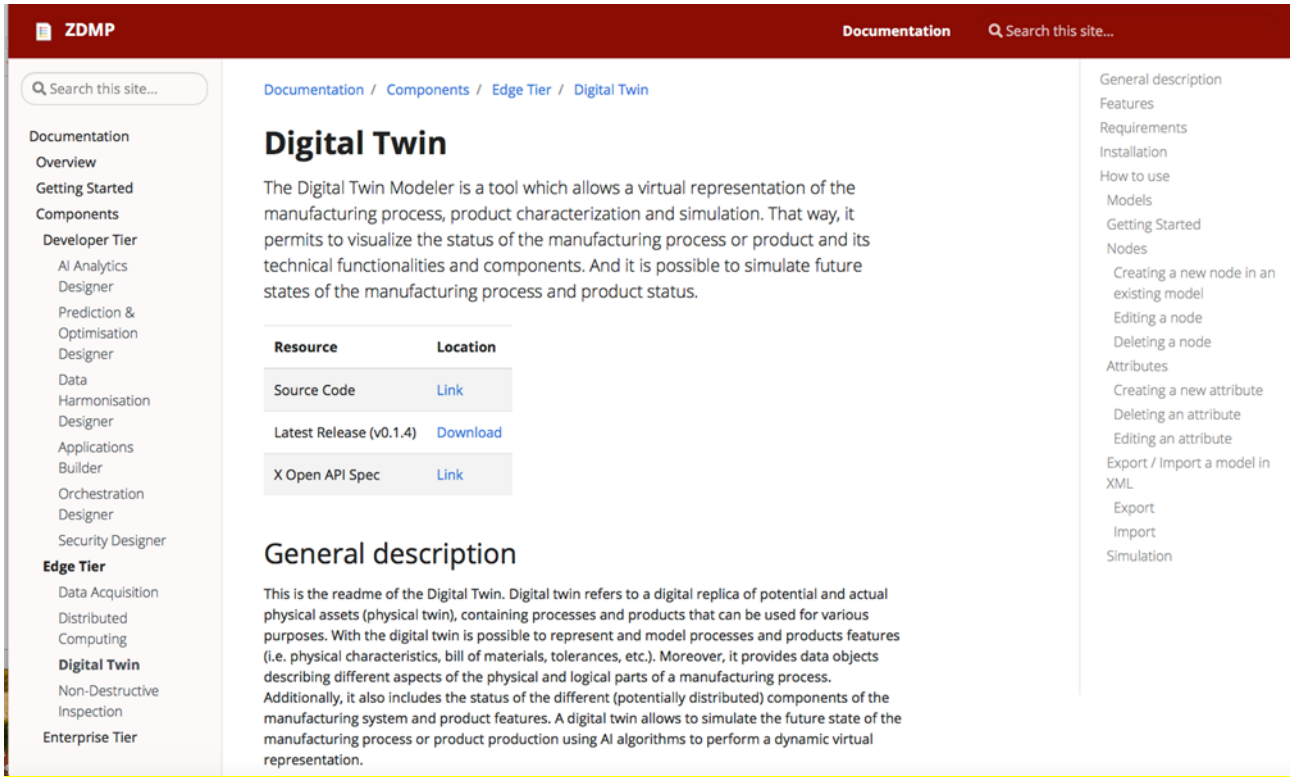
There were no issues found in this period which would limit the final scope of the activities expected.

4 T7.3 Production Stage: Material and Energy Efficiency

4.1 Public documentation

Public documentation is available as indicated in Section 1.4. Regarding this component the description is available at:

https://software.zdmp.eu/docs/components/edge-tier/digital-twin/



CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/edge-tier/t7.3-t8.1-digital-twin
Latest release:	Available	Vs 1.0.0
Open API Spec:	Available	This section is being updated periodically as new functionalities are being developed
General Description:	Available	N/A
Features:	Available	N/A
System Requirements:	Available	N/A
Installation:	Available	N/A
How to Use:	Available	This section is being updated periodically as new functionalities are being developed

4.2 M18 Report

4.2.1 Architecture Implementation

The first version of the Digital Twin (DT) component employs a licensed time-series database and a model database on the background, both based on the licensed System PI Infrastructure, which allow progressing on the developments in this first version. After M18, the connection with open source databases to store the models and time series (signals), will be developed using Ditto Framework or/and Mongo DB database system and integrating also with the Storage and Bus components.
The first version of WEB user interface for the Digital Twin Modeler is a read-only mode structure of a DT model already existing. Concretely, this first version includes: functionalities of editing/ creating elements (nodes), graphic trends for dynamic attributes, the user interface to edit/create simulation nodes and the first version of functionalities for WEB API Digital Twin component (import model database, export database, read from a node and read from direct attribute), based on property formats.
Figure 5 below shows the status of the architecture implementation:

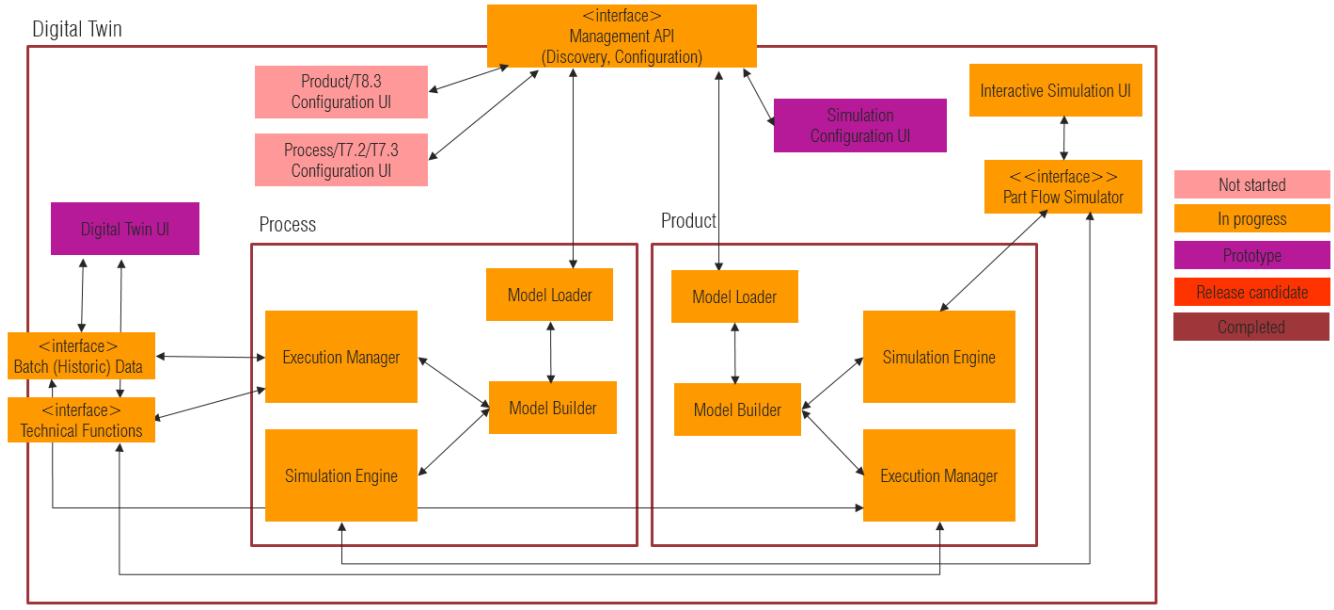


Figure 5: Digital Twin architecture

4.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Digital Twin Modeler User Interface able to show a basic Digital Twin Model already created:** Visualization of a simple Digital Twin Model already created in read-only mode, in front of a System PI infrastructure on the background
- **[Completed] Tree Hierarchy Navigation:** Development of a tree hierarchy navigation of the nodes of a Digital Twin sample, based on the read-only mode in front of a sample of a Digital Twin model stored in a System PI infrastructure
- **[Completed] Development of visualization of datasheets of assets:** Development of Digital Twin Modeler User Interface allowing to show datasheets of assets in order to show attributes and dynamic values (physical signals) versus a System PI infrastructure

- **[Completed] Digital Twin import/export functionality:** Development of the Digital Twin import/export functionality to an XML format versus a System PI Digital Twin infrastructure. The UI Digital Twin Modeler must be able to import/export to an XML format a Digital Twin model, or to transfer it to other components. After M18 is planned the creation of the open XML model format to export/import
- **[Completed] Mapping functionality to store signals on System PI time-series database:** Development of a mapping functionality to store signals on System PI time-series database in read-only mode. After the M18 the development of the connection with Ditto Framework storage and the ZDMP STORAGE component will be scheduled
- **[Completed] Reading Digital Twin models on Digital Twin Storage System PI database:** Digital Twin Modeler will be able to read the Digital Twin models available on the Digital Twin Storage System PI database. After M18 the development of reading Ditto Framework stored models will be scheduled
- **[Completed] Creation of a basic industrial plant of a Digital Twin sample structure**: Creation of a simple industrial plant of a Digital Twin sample structure in a System PI infrastructure to be shown in the read-only mode inside the Digital Twin Modeler UI interface
- **[Completed] Development of a simple simulation component:** The development of a basic simulation component which can create a backend service on top of a simulation model. This subcomponent has been merged and connected with T7.1/T7.2/T7.3 Prediction and Optimisation Run-time. The Modeler user interface prepares the map of signals as input and output and also creates the configuration file to connect, through WEB API, with the T7.1/T7.2/T7.3 Prediction and Optimisation Run-time subcomponent
- **[Completed] Integration simulation controller backend to Digital Twin Modeler:** Development of the integration of the simulation controller backend in the Digital Twin Modeler to allow mapping of simulation models to physical assets. The Digital Twin modeler UI allows the configuration of the input and output attributes, the mathematical simulation run, and the frequency of the calculations
- **[Ongoing] Development of a simulation controller to execute simulation scripts:** Development of a simulation controller to execute simulation scripts according to the set of user-defined rules (eg signals every 10 seconds) and to publish the simulation results in the messaging and storage subcomponents. This subcomponent has been merged and connected with T7.1/T7.2/T7.3 Prediction and Optimisation Run-time subcomponent. The Modeler user interface is able to prepare the map of signals as input and output and the configuration to connect, via WEB API, with T7.1/T7.2/T7.3 Prediction and Optimisation Run-time, sent as part of the model. The result is stored on a preliminary database version at this moment

4.2.3 Progress

See Section 2.2.3 for comments on the methodology.

Sprint 1 (Jan 27, 2020 – Feb 9, 2020)
• **Frontend Digital Twin Modeler viewer v0.2 VS System PI (from previous work oct-19 to jan-20).** Angular App that implements read-only visualization of nodes and attributes coming from a Digital Twin model already created. v0.1 Digital Twin MODELLER user interface look and feel of first version design. Frontend main structure of the first version. Prepare Frontend base angular structure. Related Subtask: DT006 • **Backend Digital Twin Modeler viewer v0.2 VS System PI (from previous work oct-19 to jan-20**): Initial version of a REST API vs System PI Web API. V0.1.This version implements read-only

retrieval of node and attribute information from OSI Soft PI Web API of a Digital Twin model already created. Related Subtask: DT003, DT004

- **Real-time System Cloud Infrastructure on CET servers (work coming from oct-19 to January-20).** Setup servers, Installation and configuration of System PI Software (Data Archive, Asset Framework, PI Web API/PI Vision)... Virtual servers installation on CET cloud platform to run AF WEB API, RT database, SQL Database, and RT Analysis RT in order to be used to be connected with the Digital Twin modeler to be developed
- **Backend development (split in minor issues).** Improve communication with PI Web API via REST. Related Subtask: DT003, DT004
- **Defining and preparing architecture for developing of Digital Twin Modeler (from previous work (oct-19 to january-20).** Defining and preparing Architecture and management for developing the Digital Twin Modeler API backend and front end. Starting to develop a first common structure. Defining main structure for Digital Twin Modeler Developing

Sprint 2 (Feb 10, 2020 – Feb 23, 2020)

- **Investigating how to interconnect and prepare Digital Twin components and Modeller with equipment performance through especial Optimisation elements on the Digital Twin Modeller.** Optimized Signal (support calculations) - Inputs: Other attributes - Outputs: New attribute/s mapped to signals to bus or message and storage as OPTIMIZED ATTRIBUTE - Must be selected: Mathematical procedure STORED on SIMULATION ENGINE - Must be selected: Periodicity of calculation for the ENGINE SIMULATION EXECUTION ENGINE. This structure aproach will be developed on future upagrades. Related Subtask: DT005
- **Meeting with FORD for technologies review with ITI.** Meeting with FORD for technologies review with ITI in order to know what is possible to implement in the user cases
- **Build a first version of functional story about zApp Digital Twin.** Contributing to the first versión of functional story for scenario prepared for Profactor
- **Differentiate between static attribute and signal attribute (dynamic).** Front end capability for Digital Twin Modeller to differentiate between static attribute and dynamic attribute (signal) by differentiating visualization reading from System PI database. Related Subtask: DT006

Sprint 3 (Feb 24, 2020 – Mar 8, 2020)

- **Research on how to interconnect prediction on process quality trough especial prediction elements on Digital Twin component.** As conclusion is proposed to have a Prediction Signal (Future prediction) with this structure: - Inputs: Other attributes -Outputs: New prediction attribute/s mapped to signals to bus or message and storage - Must be selected: Mathematical procedure STORED on SIMULATION ENGINE - Must be selected: Periodicity of calculation for the ENGINE SIMULATION EXECUTION ENGINE. This structure approach will be developed on future developing upgrades. Related Subtask: DT005
- **First list version of databases needed for Digital Twin on storage component.** First list of databases needed for Digital Twin is sent to storage component leader.Ditto Framework and MongoDb are proposed
- **Defining Top 10 issues functionalities for M18.** Defining the Top 10 issues functionalities expected for M18 for the Digital Twin component
- **Retrieval of the static attribute from test database System PI (backend).** Developing the backend capability to retrieve static values from test System PI database to be shown on Digital Twin Modeller User Interface. Related Subtasks: DT003, DT004
- **Prepare PPT first version of features expected for Digital Twin UI Modeller on 2021.** Preparing power point with the global features expected for Digital Twin UI Modeller and component
- **Nodes Tree view visualization improvement of frontend vs System PI.** Developing an improvement of frontend nodes tree view visualization of Digital Twin UI Modeller for improved visualization and speed unfolding and collapsing the navigation tree view nodes. The model is read from System PI database. Related Subtask: DT006

Sprint 4 (Mar 9, 2020 – Mar 22, 2020)

- **Pre-Meeting with partners to show PPT with the main idea of features of Digital Twin Modeler version for 2021.** Explanation and discussion of the functionalities and features of Digital Twin Modeler version for 2021 and implemented features expected on M18
- **Frontend design with buttons for the main attributes types.** Design in Digital Twin Modeler User Interface the buttons layout for attributtes(static, dynamic, simulation, etc). Not functional. Related Subtask: DT006

- **Backend web API capability to organize nodes in a hierarchical tree structure vs System PI.** Development of the Digital Twin API functionality to shown and navigate through a hierarchical tree structure of nodes. The model is read from System PI database. Related Subtasks: DT003
- **Backend capability to read the current value of a dynamic attribute vs System PI.** Development of the Digital Twin API functionality to read the current value of a dynamic attribute (real time signal). The model is read from System PI database. Related Subtasks: DT003, DT004
- **Backend capability to read the value of a static attribute vs System PI.** Development of the Digital Twin API functionality to read the value of a static attribute. The model is read from System PI database. Related Subtasks: DT003, DT004
- **Docker compose installation and testing.** Installation of the Docker Toolbox software in virtual machine Windows. Installation of Docker Desktop sotware in virtual machine, testing and detection of bug in Docker Desktop which does not work on virtual machines. Testing Docker Toobox

Sprint 5 (Mar 23, 2020 – Apr 5, 2020)

- **Meeting with WP8 and WP7 leaders and UPV and ITI to show and discuss about the features of Digital Twin.** New discussion about features of first digital twin version for 2021 and implemented features expected on M18
- **Create a first model example of an industrial plant.** Create the model in a System PI database, with real time simulated signals. This model will be used in the next step as a model to be shown (read only) in Digital Twin Modeler User Interface
- **Frontend capability to organize and navigate nodes in a hierarchical tree structure vs System PI.** Development of the Digital Twin Modeler User Interface capability to shown and navigate through a hierarchical tree structure of nodes. The model is read from System PI database. Related Subtasks: DT003, DT006
- **Frontend capability to read the current value of a dynamic attribute vs System PI.** Development of the Digital Twin Modeler User Interface capability to read the current value of a dynamic attribute (real time signal). The model is read from System PI database. Related Subtasks: DT003, DT004, DT006
- **Frontend capability to read the value of a static attribute vs System PI.** Development of the Digital Twin Modeler User Interface capability to read the value of a static attribute. The model is read from System PI database. Related Subtasks: DT003, DT004, DT006

Sprint 6 (Apr 6, 2020 – Apr 19, 2020)

- **Backend capability to show a simple model already created vs System PI.** Development of the Digital Twin API functionality to show an existing model. The model is read from System PI database. Related Subtasks: DT003, DT004
- **Frontend capability to show a simple model already created vs System PI.** Development of the Digital Twin Modeler User Interface capability to show an existing model. Visualization on read-only mode. The model is read from System PI database. Related Subtasks: DT003, DT004, DT006
- **Backend capability to show a trend of a dynamic attribute vs System PI.** Development of the Digital Twin API functionality to show the trend of a real time signal (dynamic attribute). The model is read from System PI database. Related Subtasks: DT006
- **Backend capability of exporting a Digital Twin model to XML file vs System PI.** Development of the Digital Twin API functionality to export a model from Digital Twin to a XML file. The model is read from System PI database. Related Subtasks: DT003
- **Investigate graphics libraries.** Investigate graphics libraries to show dynamic data and signals results. Related Subtasks: DT006

Sprint 7 (Apr 20, 2020 – May 3, 2020)

- **Study how to implement Element Simulation on Digital Twin Modeler and how to integrate with UPV first version simulation subcomponent.** After a meeting with UPV it is decided to connect with run time T7.1-T7.2-T7.3-Prediction and Optimization Run-time through API. Related Subtasks: DT005
- **Backend capability to show a node datasheet.** Development of the Digital Twin API functionality to show the information of the static and dynamic attributes contained in a node. The model is read from System PI database. Related Subtasks: DT003
- **Frontend capability to show a node datasheet.** Development of the Digital Twin Modeler User Interface capability to show the information of the static and dynamic attributes contained in a node. The model is read from System PI database**.** Related Subtasks: DT003, DT004, DT006

- **Frontend Capability of exporting a Digital Twin model to XML file vs System PI.** Development of the Digital Twin Modeler User Interface capability to export a model from Digital Twin to a XML file. The model is read from System PI database. Related Subtasks: DT003, DT006
- **Frontend capability to show a trend of a dynamic attribute vs System PI.** Development of the Digital Twin Modeler User Interface capability to show the trend of a real time signal (dynamic attribute). The model is read from System PI database. Related Subtasks: DT005

Sprint 8 (May 4, 2020 – May 17, 2020)

- **First version of documentation on GitLab.** Preparation of a first version of Digital Twin documentation on Gitlab
- **Frontend capability to import a XML file of a model vs System PI.** Development of the Digital Twin Modeler User Interface capability to import an existing model in XML format and upload it in the Digital Twin Modeler. The model is uploaded in System PI database. Related Subtasks: DT003, DT006
- **Frontend capability to map input and otput attributes for a simulation.** Organizing the parameters needed in a simulation**.** Related Subtasks: DT005, DT006
- **Frontend for Digital Twin Modeler to be able to call the simulation controller backend in the Digital Twin Modeler to allow mapping simulation models to physical assets.** Finally connecting with WEB API of T7.1-T7.2-T7.3-Prediction and Optimization Run-time. Related Subtasks: DT005, DT006
- **Backend capability to import a XML file of a model vs System PI.** Development of the Digital Twin API functionality to import an existing model in XML format and upload it in the Digital Twin Modeler. The model is uploaded in System PI database**.** Related Subtask: DT003
- **Frontend capability** to **to select from several models available on Digital Twin storage vs System PI.** Development of the Digital Twin Modeler User Interface capability to show and select a model from the Digital Twin model storage. The model is read from System PI database. Related Subtasks: DT003, DT006
- **Backend capability to select from several models available on Digital Twin storage vs System PI.** Development of the Digital Twin API functionality to to show and select a model from the Digital Twin model storage. The model is read from System PI database. Related Subtask: DT003

4.2.4 Limitations

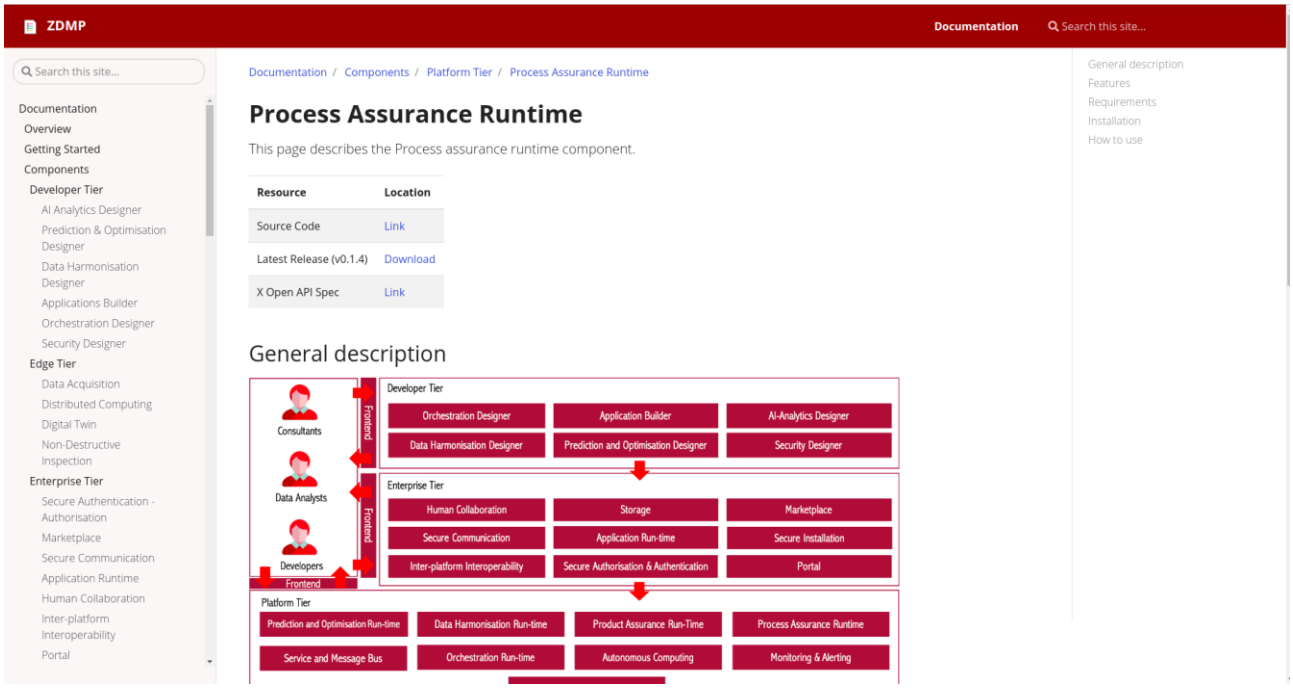
There were no issues found in this period which would limit the final scope of the activities expected.

5 T7.4 Process Quality Assurance

5.1 Public documentation

Whilst some light activity has taken place, more intensive development of this component is planned to start after sprint 9 of PO Run-time. Hence, at this point a significant online documentation cannot be delivered for this component. Public documentation is available as indicated in Section 1.4. Regarding this component the description is available at:

https://software.zdmp.eu/docs/components/platform-tier/process-assurance-runtime/



CURRENT on-line documentation status/exceptions:

Content	Status	Explanation
Source code:	Available	https://zdmp-gitlab.ascora.eu/zdmp_code/platform-tier/t7.4-process-assurance-run-time
Latest release:	Available	Vs 0.4.0
Open API Spec:	N/A	N/A
General Description:	N/A	N/A
Features:	N/A	N/A
System Requirements:	N/A	N/A
Installation:	N/A	N/A
How to Use:	N/A	N/A

5.2 M18 Report

5.2.1 Architecture Implementation

The component Process Assurance Run-time (PA Run-time), acts as a hub that utilizes the other components developed in ZDMP platform, mostly “prediction and optimisation Run-time” (PO Run-time). PA Run-time will use PO Run-time as a numerical processor

and delegates to it the operations such as prediction of future process values or quality measures, optimisation, and recommendation.

Since PO Run-time (T7.2) has been the fundamental component for PA Run-time (eg PA depends on PO Run-time), its development has been prioritized to development of PA Run-time (T7.4). In fact, this dependency extends beyond PO Run-time, to other components such as Digital Twin and Alerting and notification components.

Figure 6 below shows the status of the architecture implementation:

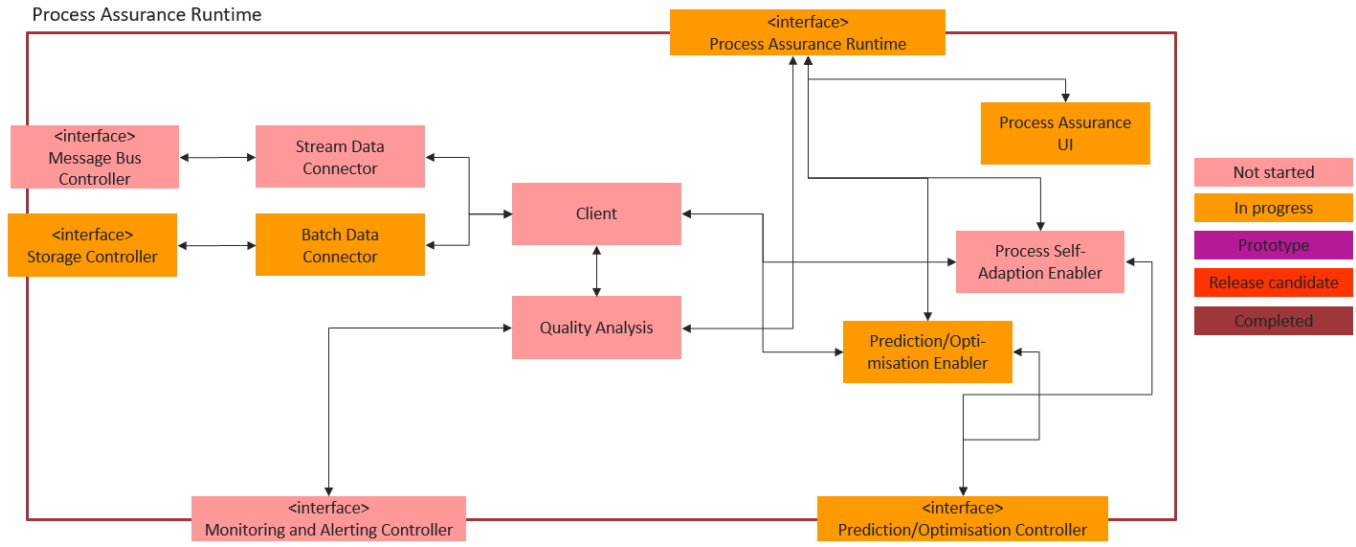


Figure 6: Process Assurance Run-time module diagram

5.2.2 Planned Activities

These were the planned activities for this period with an overview of their status:

- **[Completed] Backend Skeleton:** Define a docker container for an uWSGI Server that uses a Flask App
- **[Completed] Frontend Architecture:** Define a docker container with a Nginx Server that serves a React UI
- **[Completed] Setup connection to new PO Run-time:** Connects to a PO Run-time sends input and receives the prediction result
- **[Completed] Integrate UI from a connected PO Run-times:** Recreate the UI environment form PO Run-time
- **[Completed] Trigger prediction/Optimisation of connected PO Run-time:** send a prediction/optimisation request to PO Run-time including the setup info
- **[Ongoing] Define API:** Define the Open API Specification of the Python backend
- **[Ongoing] Display result of connected PO Run-time:** Show plot of a predicted value
- **[Not started] Process Assurance UI:** To setup configuration of PO Run-time (input/output config), setup thresholds, and execute the scenario (eg start a PO Run-time, request prediction and plot and monitor the result)
- **[Not started] Setup responses for events:** Eg trigger an alert if a prediction is below a certain value. This response can be in connection with Digital Twin. Action can be changing a property of an asset, for example, turning a light or valve on/off.
- **[Not started] Specify a data source for a connected PO Run-time:** The user selected data source will be sent to the PO Run-time

5.2.3 Progress

See Section 2.2.3 for comments on the methodology.

As explained in the beginning of this section, development of PO runtime (T7.2) is prioritized to this component. For this reason, the activity and number of addressed issues in the following tables are minimal.

Sprint 1 (Jan 27, 2020 – Feb 9, 2020)
• **Define main contributor from CET.** Discussion on TODO's list and issues for the component • **Define main contributor from PROF.** Discussion on TODO's lists and issues for the component • **Assign a main developer.** IKER has to assign a main developer person to the component
Sprint 2 (Feb 10, 2020 – Feb 23, 2020)
• **Studying with WP partners about the contribution to the shared sub-components.** Discussion about the implementation of the component and the role of each partner
Sprint 3 (Feb 24, 2020 – Mar 8, 2020)
• **Data selection for quality analysis.** Discussions about how and which synthetic data can be used, due to the lack of real-time data
Sprint 4 (Mar 9, 2020 – Mar 22, 2020)
• **Meeting to define the contribution of partners to each component.** Telco to define the contribution and the best way to implement the work in the component. Related Subtask: T74A008
Sprint 5 (Mar 23, 2020 –Apr 5, 2020)
• **Documentation.** Fixes to online documentation material. Additional documentation materials and demo environment for plenary. Related Subtask: T74A008 • **Plenary Technical Meeting**
Sprint 6 (Apr 6, 2020 – Apr 19, 2020)
• **Use a DB to store state (eg PO configs) in the backend.** To persist the configuration of the connected PO runtime save them in a DB. Related Subtask: T74A007 • **Add (by specifying the URL) and configure PO-Run-times using the UI.** The List of URIs of connected/known PO-Run-times is stored in the backend - Configuration is done via the frontend. Related Subtask: T74A001 • **Trigger prediction/Optimisation of connected PO Run-time.** Send a prediction/Optimisation request to PO Run-time including the setup info. Related Subtask: T74A008 • **Integrate UI from a connected PO Run-times.** Recreate the UI environment from PO Run-time. Related Subtask: T74A004 • **Setup connection to new PO Run-time.** Connects to a PO Run-time sends input and receives the prediction result. Related Subtask: T74A008 • **Frontend Architecture.** Define a docker container with a Nginx Server that serves a React UI • **Backend Skeleton.** Define a docker container for an uWSGI Server that uses a Flask App
Sprint 7 (Apr 20, 2020 – May8, 2020)
• **Adapt UI design:** Adapt the UI that is moved from PO runtime to this component accordingly. Related Subtask: T74A008

5.2.4 Limitations

There were no issues found in this period which would limit the final scope of the activities expected.

6 Risks

The risk status associated with this WP and underlying tasks are as follows:

Risk	WP Likelihood % Impact 💣	Critical risks for implementation - Proposed risk-mitigation measures -	Current Status
Failure in Functions (ZD production and products)	WP: WP7/8 %: Low 💣: High	This is considered as a core issue in the project. This is marked low occurrence due to the previous research that has been performed and the success of the dependent projects; Nonetheless, having multiple pilots in the project will allow early detection of the problem if it occurs and WP11 will work as early testing environment to avoid such issues.	This task has not started in earnest; thus, no change to risk level **Risk future change:** % Low: No change 💣High: No change
Available datasets	WP: WP7/8 %: Low 💣: High	This is considered as a core issue in the project and particularly this WP. This is marked low occurrence due to the availability of multiple use cases; nonetheless, early detection of any potential problem must be monitored and WP11 will work as early testing environment to avoid such issues.	Monitoring **Risk future change:** % Medium: No change 💣 High: No change
User Cases datasets not enough relevant to make predictions	WP: WP7 %: Medium 💣: Low	Affecting mainly to R&D partners (PROFACTOR, UPV), the datasets provided by user cases could maybe do not have relevant information in order to create predictive models. After pre-analysis could be necessary to ask for new datasets to customers.	Analysis of user cases datasets not started Risk future change: % Medium: No change 💣High: No change
WP7/8 Overlapping	WP: 7/8 %: Low 💣: High	There is a risk that WP7 and WP8 partners may develop the same functionalities in parallel if there is no close cooperation. The project needs a systematic approach to clearly differentiate activities in both work-packages from a functional perspective	The WP7/8 scientific leaders are jointly working on a methodology to clearly define the functional blocks developed in each WP, identify synergies and cooperate closely to avoid effort duplication. Risk future change: % Medium: No change 💣High: No change

7 Conclusions

This document reports on the different WP7 tasks' activities and components development providing the status at M18.

It first illustrates ZDMP WP7 work in the scope of ZDMP global architecture and describes what tasks and components compose it. Its main aim is to create tools for the users of the platform that provides process quality assurance using prediction and optimisation techniques along the entire process chain.

It is composed of the following tasks/components:

- T7.1/T7.2/T7.3/T7.4 Prediction and Optimisation Designer
- T7.1/T7.2/T7.3 Prediction and Optimisation Run-time
- T7.3 Digital Twin
- T7.4 Process Assurance Run-time

All these tasks/components are monitored and evaluated against the requirements presented in D048 Requirement Document and Update, and Software development approach is based on agile sprints following D021 Methodology Document.

Every section of the document describes the achievements of each task/component and is being complemented by the software reports found in the public repository.

The main challenges in this first iteration were the definition of the components, the linking of the requirements from the users to the technologies of the partners in WP7 and the distribution of responsibilities amongst partners. However, the consortium put in place the following measures to overcome these challenges:

- Excel list with linking of the WP7 Tasks to the use cases
- Workshop during plenary meeting in Bremen to develop the architecture of the components
- Harmonisation with WP8 to discover any overlapping in development (eg Digital Twin)
- Bi-weekly Telco's to discuss and monitor progress in the sprints for every component

The main development of this period were software components for the specific tasks in WP7. These components will serve as a good basis for the further evolvement of the software and the mathematical algorithms which are standing behind it. The partners have built a good team after this first period and are working together smoothly and with a good open communication.

Annex A: History

Document History	
Versions	V0.0.1 • First tentative TOC and initial content in section 0 V0.0.2 • Draft content for T8.2-T8.4 Product Quality Assurance section V0.0.3-4 • New drafts after coordinator comments V0.0.5 • Input from all WP7 partners V0.0.6 • Review V0.0.7 • Corrections V0.0.8 • Review V1.0.0 • Corrections and first release V1.0.1 • Minor changes and corrections V1.1.0 • PM review and submitted version V1.1.0A • PM review and submitted version
Contributions	PROF: • Daniela Kirchberger • Kastor Felsner • Amirreza Baghbanpourasl UPV: • Francisco Fraile CET • Ernesto Bedrina IKER • Ana Gomez ITI: • Francisco Barrena • Santiago Cáceres ICE: • Stuart Campbell

Annex B: References

None

ZERO DEFECTS
Manufacturing
Platform
ZDMP

www.zdmp.eu