H2020-EINFRA-2017
EINFRA-21-2017 - Platform-driven e-infrastructure innovation
DARE [777413] "Delivering Agile Research Excellence on European e-Infrastructures"

Decomposed User Stories and Tooling

Project Reference No	777413 — DARE — H2020-EINFRA-2017 / EINFRA-21-2017
Deliverable	D3.1 Decomposed User Stories and Tooling
Work package	WP3: Large-scale Lineage and Process Management
Tasks involved	T 3.1: Decomposed User Stories
Type	R: Document, report
Dissemination Level	PU = Public
Due Date	30/06/2019 Deadline extended to 31/07/2019 in agreement with PO
Submission Date	31/07/2019
Status	Final Draft
Editor(s)	Rosa Filgueira (UEDIN), Alessandro Spinuso (KNMI)
Contributor(s)	Malcolm Atkinson, Amy Krause (UEDIN), Andrea Pagani (KNMI), Federica Magnoni (INGV), Iraklis Klampanos, Antonis Koukourikos, Aggelos Charalambidis (NCSR-D)
Reviewer(s)	Antonis Koukourikos (NCSR-D)
Document description	Overview of the Agile User Stories methodology and tooling adopted. Analysis of user stories extracted from the community use cases, leading to DARE general features.

Document Revision History

Version	Date	Modifications Introduced Modification Reason	Modified by
0.1	01/07/2019	First Draft from Deliverable	Rosa Filgueira
0.3	10/07/2019	Added Features section	Alessandro Spinuso
0.5	10/07/2019	Updated Section 3	Rosa Filgueira
0.6	11/07/2019	Updated References	Alessandro Spinuso
0.7	16/07/2019	Updated Sections 3, 5	Rosa Filgueira
0.9	23/07/2019	Applied changes after Revision, Sections 2.3 and 6	Alessandro Spinuso
1.0	26/07/2019	Getting final links in the references	Rosa Filgueira

Executive Summary

The deliverable's objectives are to determine the User Stories for the Seismological (WP6) and Climate (WP7) Uses Cases from which we can conclude components and technologies required to extend their capabilities in handling the scale of data or the computational challenges, so that the DARE Platform can deliver them through a new framework and design pattern for many future applications.

We have identified a different level of maturity and point of view between the two communities. The seismological users approached the platform by looking at new functionalities that need to be implemented for improving the simulation of the geological plate movements and their effects. These users focused on functionalities and less on the underlying technical details. The climate science users looked at the user stories from a different perspective: the functionalities needed were not particularly new or advanced (i.e., computation of mean and standard deviations for a set of climate variables), but the key challenges are the performance and the distribution of computations and data sets. Such difference is reflected in the stories detailed in this deliverable. However, these user stories have helped us to identify several generic components, such as cataloguing, processing, provenance or data store, for which the DARE platform will supply new methods to support both communities' developers by exploiting different technologies.

Table of Contents

List of Terms and Abbreviations

Abbreviation	Definition
RA	Rapid Ground Motion Assessment
SS	Seismic Source Characterisation
ES	Ensemble Simulations
ACS	Analysis of Climate simulations on-demand
E	Epic (Community Use Cases)
T	Theme (Community Use Cases)
C	Component (DARE Platform)
CA	Capability (DARE Platform)
F	Feature (DARE Platform)
B2DROP	EUDAT data exchange service for researchers

1 Introduction

This deliverable is the document that is meant to crystalize the requirement of two classes of users of the DARE platform, specifically the users in the climate science domain and the users in the solid earth science domain. In order to elicit the requirements, a modern approach has been used that takes advantage of Agile methodology by creating user stories [1]. In user stories users describe their interaction with the ideal system in high-level terms focusing on their role, their requirements for the system to perform certain tasks, and the motivation or aim of that task. A refinement process takes place to detail the stories further, iteration after iteration, to achieve a granularity that can satisfy the technical personnel that take care of the development.

In this deliverable the process and methodology are described (Section 2) and the specific stories and derived requirements for the two DARE users communities who are helping co-develop and evaluate DARE's approach are presented (Sections 3 and 4).

1.1 Purpose and Scope

This document is designed to gather the requirements that have to drive the efforts to create the DARE platform and in general to guide the focus of the DARE project. The collection of the requirements is performed by the means of user stories following the best practices of the Agile methodology [2]. User stories are a natural way to elicit the requirements that real users of the DARE platform would like to have implemented in the platform.

The focus has been limited to the two user communities key for the DARE project: computational seismologists in EPOS and climate-impact modelling in IS-ENES. However, the higher-level general requirement for a science gateway that allows scientists to perform computation in a distributed and cloud-based environment to process big data are an underlying requirement that at architectural and technical level has been always a priority goal.

1.2 Approach and relationship with other Work Packages and Deliverables

The main approach to gather the user stories is the application of the Agile methodology [1, 2]. However, in the context of DARE, such a methodology has not been applied as a dogma but has been adapted to a distributed EU project more focused on research than on pure software development (where Agile methodology originates).

For this purpose, a set of initiatives have been organised in order to facilitate the requirement collection and generation of the user stories:

- Ad hoc meetings and conference calls between WP leaders and task leaders
- A split of the user stories in two groups with two task forces focused on the respective user communities: seismology and climatology
- Use of ad hoc text forms to fill to stimulate the elicitation process with the two user communities
- Selection and tailoring of a set of tools to be used to track the user stories and their refinement process (see Section 2)
- Plenary discussion via conference call to analyze commonalities and differences between the stories generated.

Furthermore, we noted that this process of requirement gathering was a new methodology for many of the DARE team members, therefore some time was spent in emphasizing the important details of the methodology and educating the team members on how to use the Agile user stories methodology during the M6 face-to-face plenary meeting.

The elicitation of the user stories is deeply intertwined with WP6 and WP7 that detail the use cases for the seismology and climate studies, respectively. Such work packages and the respective leaders and contributors have been the major contributors to the creation of the user stories. Furthermore, a connection is present with the architecture work package (WP2), too. In the architecture the two user communities have also contributed to share their perspective, therefore also aspects of those inputs are condensed in this document. Concerning the influence of this deliverable to other work packages, the user stories are the central element around which the DARE platform (at least in the scope of the EU project) revolves. Therefore, user stories will influence many technological decisions during subsequent phases of the project. The primary influence will be through the WP4 and WP5 that provide the technological implementation. WP2 will take due note of the user stories as it develops the architecture.

2 Agile Methodology

In DARE we decided to adopt Agile Software development methods and tools to control, trace and assess progress of our development process. As a European project, DARE is characterised by different institutions. They are injecting the different aspects of the Agile framework incrementally, aiming initially at a smooth acquisition and familiarization with its governing practices. The pace of its application will be decided by developing agreement among all the technical partners, when the analysis of the Use Cases will direct our focus to well defined tasks and priorities. We will introduce in this section the basic notions of the Agile methodology and its definitions, followed by a short introduction of the adopted tools.

2.1 Agile Definitions

Agile is an umbrella term for a family of frameworks and techniques. In Figure 1 we depict a visual representation of the Scrum Framework [3], the most popular Agile process[1]. Scrum focuses on complex products where self-organising teams deliver increments of working functionality in timeboxed iterations. It consists of five main ingredients with internal cycles to monitor the actual progress of the technical implementation. These can be summarised as follows:

- **Product Backlog:** Continuous Population and rough prioritisation of **User Stories** or **Epics.** It contains all the features, requirements, desires that produce value to the user.
- **Sprint Planning:** List of **User Stories extracted from the Backlog** to be implemented and monitored within a **Sprint.** Typically, a planning is anticipated by a few **Refinement** sessions, where the **Epics** are decomposed into smaller stories with the participation of the whole team to discuss technical implementation and tasks.
- **Sprint:** Timeboxed development period interleaved by regular team updates (Daily Scrum) to keep the developers aligned on progress and to handle blocking issues.
- **Sprint Review:** Open evaluation of the achieved technical and business results, possibly discussed with the stakeholders in a demo session.
- **Sprint Retrospective:** Internal discussion on general issues aiming at improving the team's health and modus operandi. Open evaluation on the benefits and drawbacks of the events occurred during the Sprint.

[1] https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report, page 9

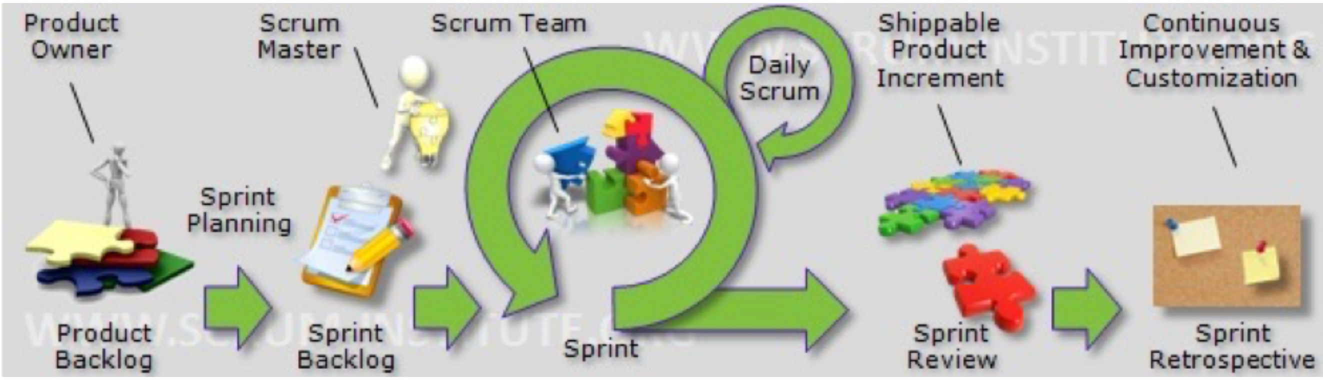


Figure 1. The Agile/Scrum process

As we can deduct from the list, the **Backlog** and its **Stories** are the most important artifacts that define and drive the dialog about the work to be done and the achievements throughout the agile process**.** Here some useful definitions to characterise the granularity of the backlog entries, their refinement and references to larger objectives:

- **User Story:** *as a <user>, I want <goal> so that <reason>*. Stimulates a series of conversations about the desired functionality. It should be user-facing, to help avoid performing technical work without direct value to a user.
- **Epic:** A larger story (or chunk of work) that has one common objective (Workflow).
- **Theme:** A scope served by more stories (or Epics), pretty much as a tag (Many Themes to Many Stories)

Alternatively, **Themes** can be also re-visited applying a perspective which focuses on the system's design:

- **Component:** A system layer (such as UI, Application or Data layer), or software module or package. An application library, or a subsystem.

Components enable features (e.g. in the case of DARE, workflow optimisation, workflow composition, workflow as a service, resource-mapping, Universe of Discourse Manager). Component's stories are typically called **Technical Stories.** Given these definitions, we can schematically organise them with also their relationships, as shown in Figure 2. Different orders of magnitude, in terms of development time, could apply as follows:

- **Theme** -> Months
- **Epic** (or Component's Features) -> Weeks
- **Story** (Tech Stories) -> Days

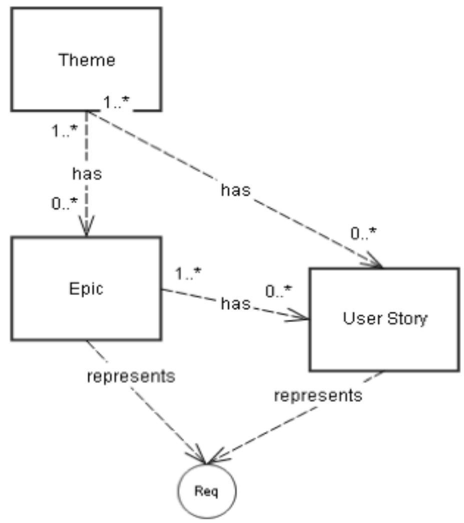


Figure 2. Agile artifacts to represent system's requirements

2.2 User Stories Best Practices

In order to proceed with the description of the Themes, Epics and Stories, we discussed a User Story cheat-list. This will help formulate the requirements at a sufficient level of granularity and precision. Initially these guidelines will help us in representing the perspective of the domain scientists at a sufficient level of granularity and precision. Eventually contributing to formulate cross-disciplinary and technical stories, which will drive the implementation of the platform's components. We list below the main principles:

- **Define & Filter:** ask 5 times why, keep cosmetics and irrelevant details out of the story, remove unknowns. Focus on the functionality and its delivered value, as perceived by the user.
- **Refine:** check the triple W: Why, Who and What. Focus on the destination (what), not on the journey (how).
- **Slicing:** Whenever a story looks too large and complex, slicing can be performed according the Divide & Conquer guidelines below.

Divide & Conquer

- **Size:** the complexity of a story should allow to get feedback as early as possible
- **Semantic split:** scanning for conjunctions such as 'and," "or," "also" and limiting phrases like "without", "unless".
- **Personas:** prioritise users first, then the user stories.
- **Divide the Flow:** identify specific steps that a user takes to accomplish a specific task.
- **Happy Path:** extract a smaller story by focusing on distinct scenarios, such as the "happy path" (main success scenario) vs. alternate (exception) flows.
- **Utility before Usability:** first make it work. Extract a smaller story by substituting basic utility for usability
- **Lower the Expectations:** extract a smaller story from the acceptance criteria to another story
- **Defer System Qualities:** move your ambitions on scalability, reliability, "to-be-more precise" to different stories (Features, tech stories of components).

- **Break Out a Spike[2]:** gain the knowledge and reduce the risk of of addressing complex stories with inadequate technological solutions (research, design, investigation, exploration, and prototyping)

Simplicity

The last section of the user stories guidelines tries to explain very basic concepts advocating simplicity in the approach to story definition and approach to evaluate technical solutions.

- **Simple Data:** extract a smaller story by focusing on a simplified dataset.
- **Simple Tasks:** extract smaller stories by focusing on simplified steps.
- **Simple Tech:** discard technologies that increase hassle, dependency, and vendor lock.
- **Batch it:** ignore access mode and decrease the granularity perceived by the end-user, if possible, by combining more subtasks.
- **Think Manual:** from simple (manual) operations to large automations.
- **Start Custom:** substituting custom for generic.

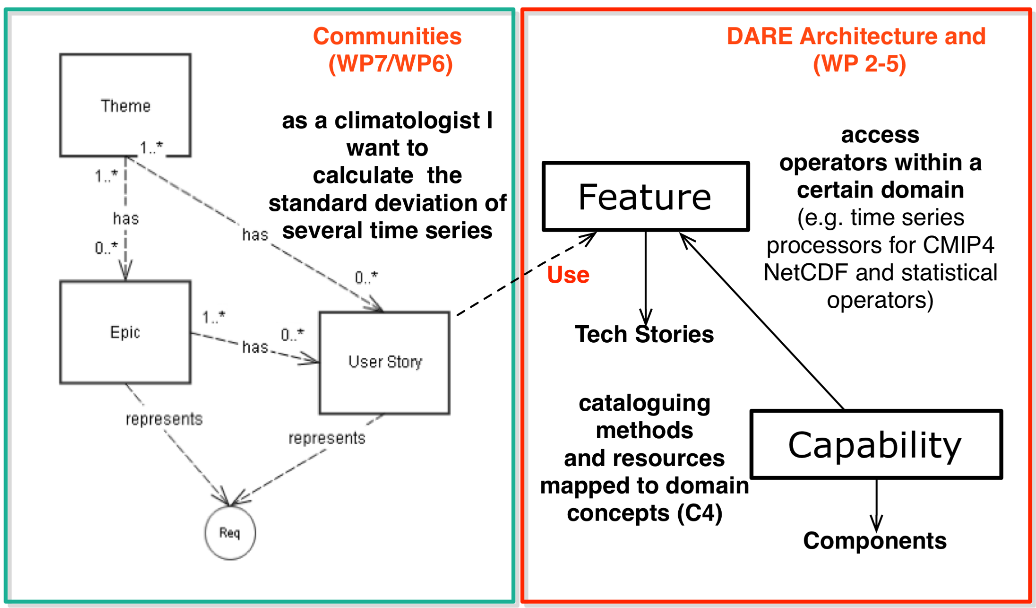


Figure 3. Mapping from Community User Stories to DARE Features and Capabilities. The user-story requires to access the right implementation of a component (Feature), which may be implemented through resolving services (Tech Story). Cataloguing is a capability taken up by WP2 and WP4.

These all very general guidelines will be taken into account during the life-time of the DARE project. However, we had to accommodate the agile process and terminology to the actual complexity of such a larger project which include stakeholders with different experience and expectations. We will explain briefly our approach in the next section.

2.3 Tuning of the Agile Process in DARE

To facilitate discussions and the understanding of the Use Cases of the two communities characterising the first user base of the DARE project (WP6 and WP7), we decided to create two groups which will

[2] Spike is a a small experiment to implement a stripped-down version of a feature, in order to figure out what needs to be built and how the team is going to build it — in advance of actually building it, to gain early feedback, potential risks, etc.

incrementally populate two separate backlogs. This approach is motivated by the need of gathering unbiased and bottom-up information about the two domains. After this first **Refinement** and prioritisation, we will proceed with mapping their requirements to the abstractions defined by the architecture (WP2), which characterises the DARE **Capabilities** (CA) and **Features** (F)**,** (see Figure 3). The iterative mapping from user stories to the DARE Architecture's model, will drive the specification of the **technical stories** that will integrate the DARE's components. These will be developed, in cooperation with WP2, WP3 and WP4, by different teams dedicated to specific set of Features. This approach will more likely comply to the Agile scenario depicted in Figure 4.

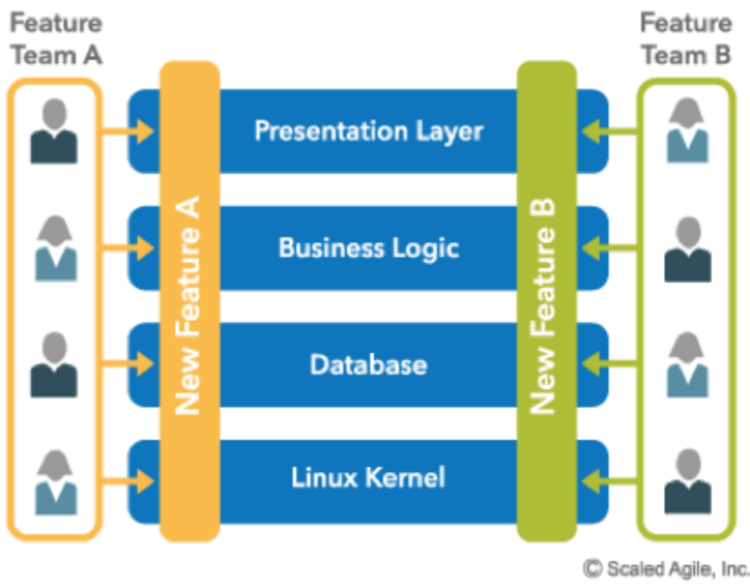


Figure 4. Components and Features Oriented Development Teams

Agile sprint-related meetings such as weekly refinements (Remote Calls) and face to face Hackathons (Athens 10/11/2018 and 24/03/2019), were scheduled in common agreement and monitored through professional tools offered by GitLab. WP1 maintains the DARE account on the official online GitLab web pages (https://gitlab.com/project-dare). Each Hackathon discussed and planned clear and achievable goal to progress with the implementation of the **Features** and **Capabilities** of the DARE platform. Refinements addressed the technical stories, which were iteratively validated in WP6 and WP7 through the realisation of their User-Stories. We report a selection in the User-Stories in the next Sections.

3 Seismology User Stories - WP6

The requirement elicitation and refinement of the User Stories for the Seismological Use Cases was conducted through the regular interactions with the members of WP6 and the material they produced. This consists of contributions to the D2.1 [5] Architecture Document and the compilation of Use Cases and Requirement Forms. The documents were openly discussed in dedicated calls and during face to face meetings with the community representative participating in DARE (WP6). They are inspired by those produced and adopted by the H2020 project ENVRI+, that conducted a similar requirement elicitation campaign on the generation and exploitation of provenance information by environmental research infrastructure in Europe.

3.1 Use Cases and Requirements Forms

With these forms the WP6 partners could start providing an initial decomposition and separation between technical requirements with substantial details. Requirements indicate fundamental characteristics of the system such as formats, services, data-stores, components, support of software toolkits, metadata vocabularies, etc. Use cases, instead, define relatively coarse grain actions, that can be high-level textual descriptions, list of actions or steps typically defining the interactions between a role and a system to achieve a goal. Requirements can have an impact on more use cases. The forms filled in by WP6 can be accessed online[3]. In the next section we will report the results of the analysis of the forms that highlighted recurrent tasks (e.g. hpc-simulation, time-series comparisons, parameters' statistical exploration, provenance collection), serving more types of investigation and relying on a well-defined set of data formats, data-services (FDSN),programming languages (Python) and software libraries (Obspy).

3.2 Value Centered Stories

Overall, after analysing the material, we organised the requirements and use-cases on core Agile artifacts such as Themes and Epics and we started to decompose those Epics that are associated with the Themes, as schematically represented in Figure 5. They set these as their main themes: Rapid Assessment (RA), Seismic Source Characterisation (SS) and Ensemble Simulations (ES) which are described in detail in the deliverable D2.1 [5] and D6.1 [7]

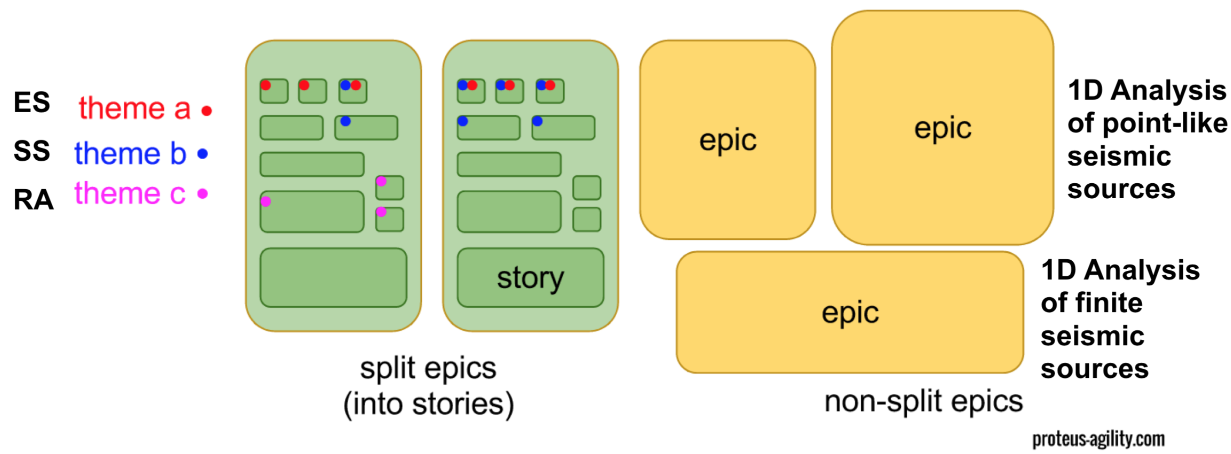


Figure 5. Schematic concept representing the division between **Themes, Epics and Stories** for the Seismology and Climatology Use Cases

The computational seismologists identified the Rapid Ground Motion Assessment (RA) Theme, whose workflow is shown in Figure 6 as their main priority. We then started analysing its details to produce an initial set of User Stories. The aim of this RA is to model the strong ground motion after large earthquakes, in order to make rapid assessment of the earthquake's impact, also in the context of emergency response (see also deliverable D6.3 [9]).

[3] https://docs.google.com/document/d/1Q9IbW1SZCskuKywX8Tz7D-PqZip7ykg4Vvn21Cq4FmE/edit?usp=sharing

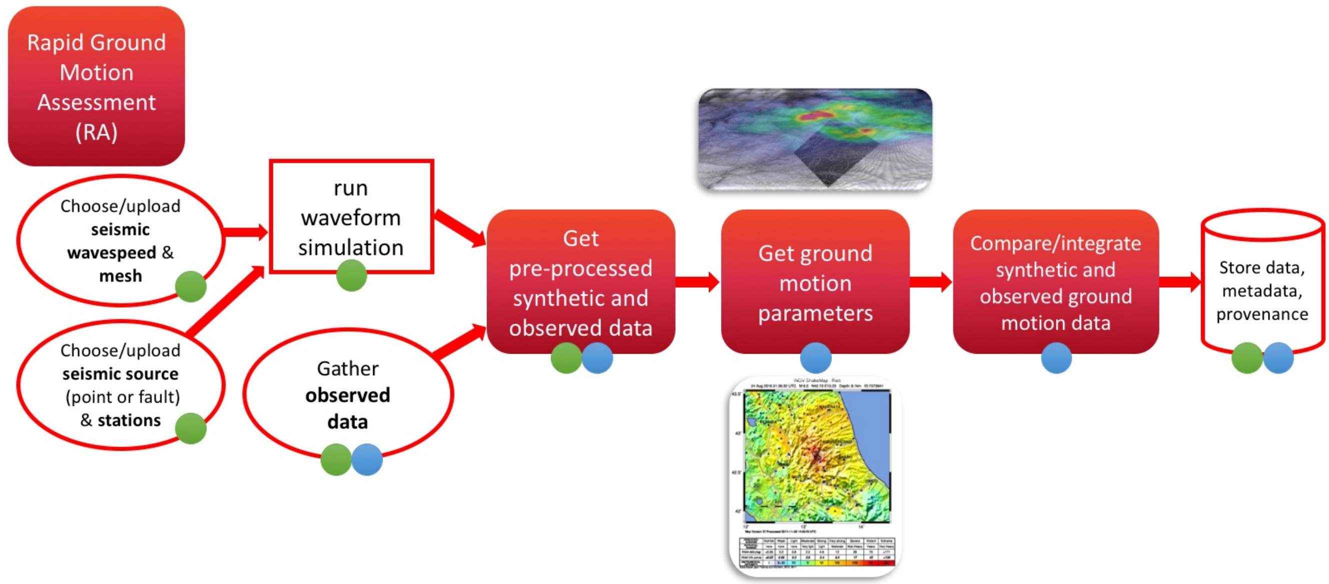


Figure 6. The Rapid Assessment (RA) Workflow and Epics, with reference to other affected Themes. green dots = Seismic Source Analysis (SS), blue dots = Ground Motion Ensemble Simulation (ES)

As it can be appreciated by reading the Use Case and Requirement gathering forms and the workflow of Figure 6, seismologists expressed the value delivered by the DARE platform in terms of very specific tasks. These are defined with enough details, offering in such a way the background that will allow the technical teams to approach the design and integration of the DARE components with a clear focus and validation targets. Figure 6 summarises these steps, extensively described in deliverable D6.1:

- Users have to choose a model to describe the physical properties and geometry of the wave propagation medium, i.e. a **wavespeed model and a mesh**.
- A model of the **seismic source** that represents the earthquake should be also defined as well as a list of seismic stations to be used in the analyses.
- With the previous inputs a **waveform simulatio**n can be run to obtain the synthetic seismograms; in the implemented example the chosen numerical algorithm/procedure is SPECFEM3D_Cartesian [*Peter et al.*, 2011], specific for local seismic waveform simulations.
- Corresponding raw **observed seismograms** for the chosen earthquake should be downloaded e.g. from public European archives, as EIDA[4].
- A **pre-processing** is applied on both **observed data and synthetics**, that includes e.g filtering the traces or removing the instrument response from the recorded data; this prepares the seismograms for the following analyses assuring the consistency between data and synthetics.
- Chosen **ground motion parameters** are calculated from both simulated and observed wavefields.
- **Comparisons** of the **ground motion parameters** between observed data and synthetics are performed. In the simple example so far considered for RA, the determined ground motion parameters are peak values of displacement, velocity and acceleration of the ground and the

[4] http://eida.rm.ingv.it

damped spectral acceleration, then the difference between their synthetic and recorded values is calculated.

- The **output products** are finally **stored** along with their metadata and complete provenance

3.3 User Stories, Conventions and Prioritisation

We report here two refinements, one at M9 (Figure 7) and another at M18 (Figure 8) of User Stories extracted from the RA Theme. These figures show the iterative and agile process we have considered, showing how descriptions and components of user stories have changed over time. These have been defined in cooperation with WP2 and WP4.

Figure 7 shows the status of the Backlog and the preliminary list of "To Do" stories at M9. We have used "Theme" and "Component" labels (T = Theme, C = Component) to indicate the stories that are shared across more than a theme, and to list the different components that a story is composed.

Epics are just large stories that will need to be further decomposed. Thus, we treat them just as stories for which there is a shared awareness that they need to go through additional refinement (label "To Refine").

Figure 8 shows the current status of previous stories at M18 of the project. All stories of the RA use case have been completed; therefore, they are listed as "closed". Furthermore, Figure 9 gives a further description of each component used in our stories.

We can find in Deliverable D6.3 [9] the implementation and demonstration details of the Rapid Assessment (RA) stories at the DARE platform.

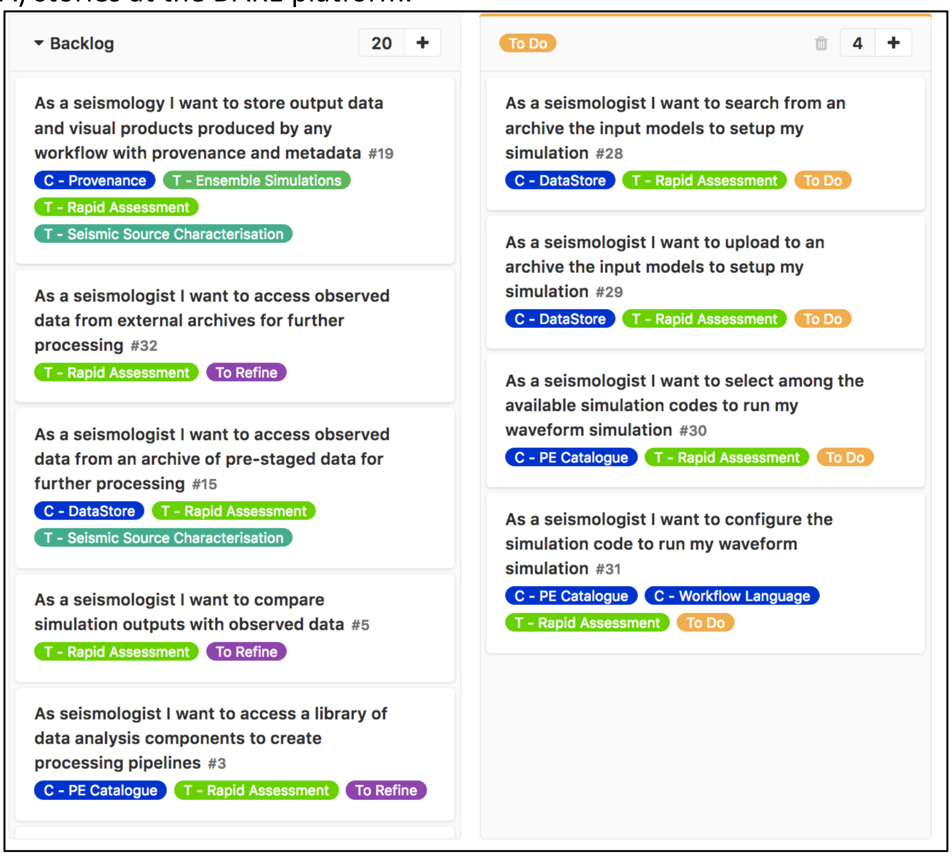


Figure 7: Backlog (at M9 of project) "To Do" list of stories to be realised. Labels with initials stand for "T" Themes, "C" Components. Stories already identified as too large are tagged "To Refine". Stories can impact on more Components and serve more Themes.

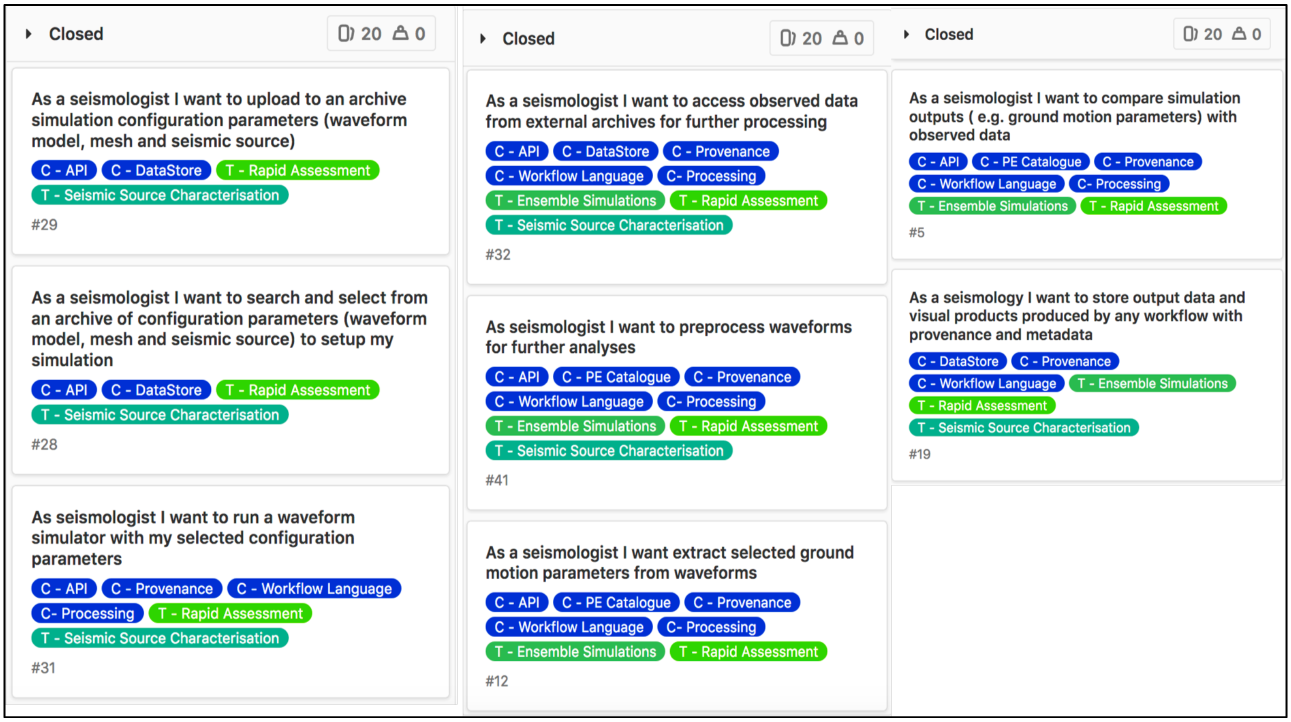
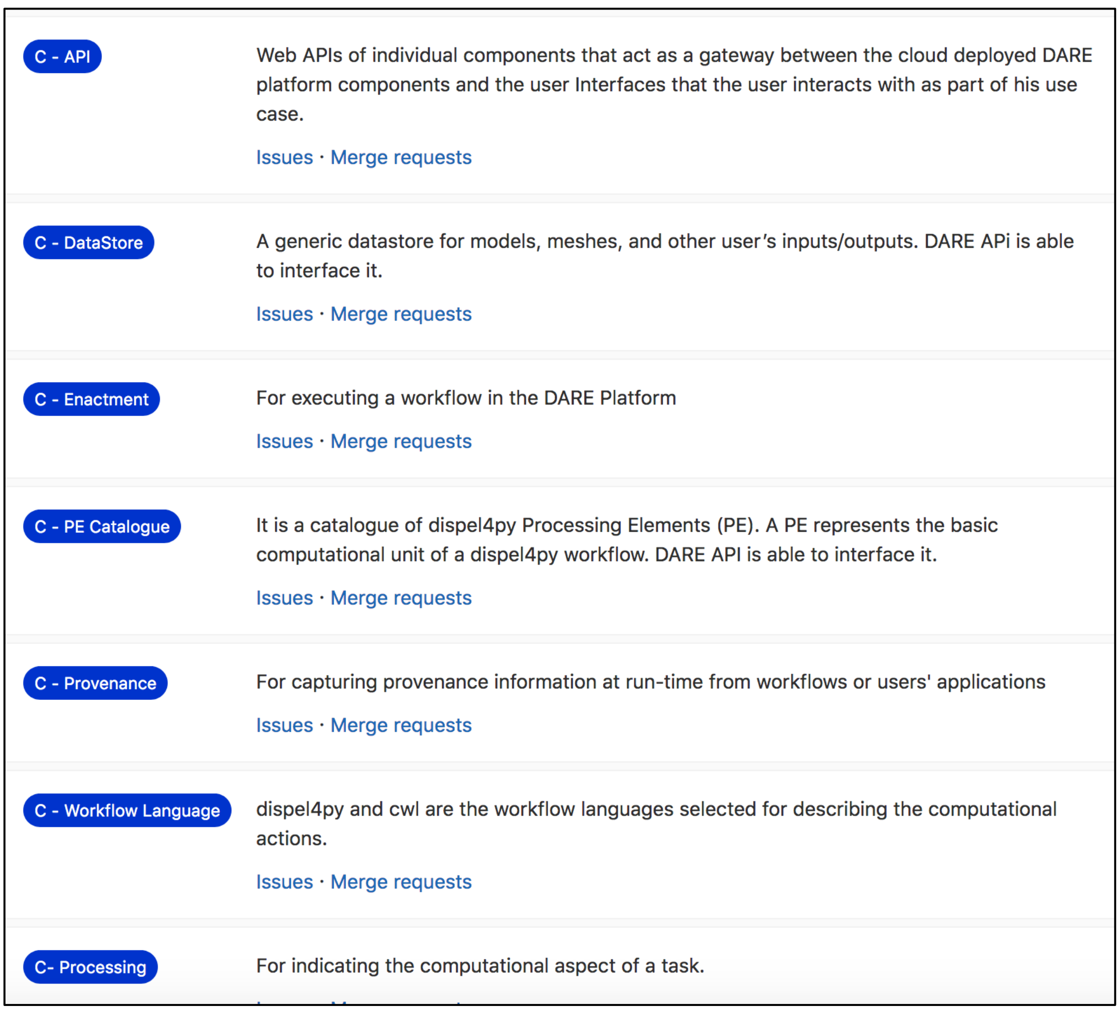


Figure 8: Current status (at M18 of the project) of the stories (all closed) for the RA use case.



C - API	Web APIs of individual components that act as a gateway between the cloud deployed DARE platform components and the user Interfaces that the user interacts with as part of his use case. Issues · Merge requests
C - DataStore	A generic datastore for models, meshes, and other user's inputs/outputs. DARE APi is able to interface it. Issues · Merge requests
C - Enactment	For executing a workflow in the DARE Platform Issues · Merge requests
C - PE Catalogue	It is a catalogue of dispel4py Processing Elements (PE). A PE represents the basic computational unit of a dispel4py workflow. DARE API is able to interface it. Issues · Merge requests
C - Provenance	For capturing provenance information at run-time from workflows or users' applications Issues · Merge requests
C - Workflow Language	dispel4py and cwl are the workflow languages selected for describing the computational actions. Issues · Merge requests
C- Processing	For indicating the computational aspect of a task.

Figure 9: Definition of the Components labels used in our stories.

4 Climate User Stories - WP7

The Climatology Uses cases was conducted through regular interactions with the members of WP7 and the material they produced. The stories were collected in two stages. The first via openly discussing the deliverables from WP2 (D2.1-M12 Architecture Document [5]) and WP7 (D7.1 IS-ENES/Climate4Impact Use Case [6]) in dedicated calls. The second, during the face-to-face meeting held in Edinburgh (12th to 14th June 2018).

4.1 User Stories, Conventions and Prioritisation

The goal of the most climate impact use-cases is to **provide to end-users data analytics on-demand** to analyze climate variables for multi-scenario, multi-model and multi-ensemble-member simulations. Therefore, we identified "**Analysis of Climate Simulations on-demand" (ACS)** as **our theme** (see Figure 10), formulating the climatology user stories to develop all the necessary components to address multiple types of analysis.

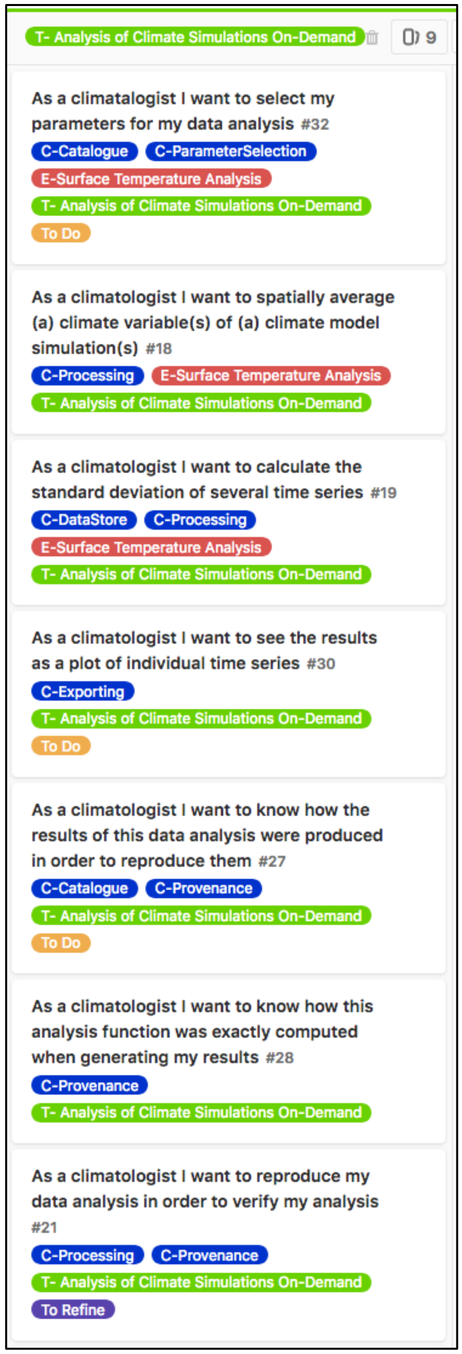


Figure 10. Labels with initials stands for "T" Themes, "C" Components. These are the stories for the **Analysis of Climate Simulations on Demand (ACS)** Theme. The figure shows seven out of the nine **ACS** stories.

However, in order to start with a specific analysis, we agreed to work on **analysing the surface temperature (ST)** during 1950-2100 time-period over Western Europe as **our epic**. The epic workflow is the following one:

- Step 1: For a given a set of parameters (for each GES scenario, for each climate model simulation and ensemble member), extract the surface temperature fields over Western Europe.
- Step 2: For each temperature field extracted, then perform a spatial average and provide as a result one time-series per temperature extracted.
- Step 3: Calculate the standard deviation of all the time series.

- Step 4: Results will be displayed on the Climate4Impact platform as a plot of individual time series, overlaying the time series average.

For capturing the previous analysis, we defined the **Surface temperature Analysis** epic: "*As a climatologist I want to generate a (multi-)model and (multi-)scenario time series average of the surface temperature*.'' We have sliced this epic into three stories (see Figure 11).

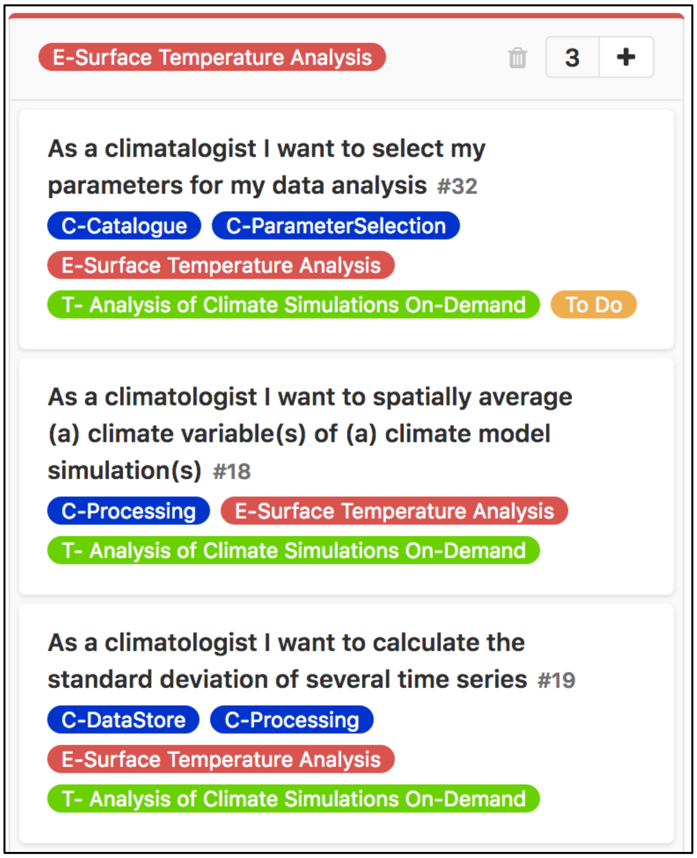


Figure 11. Decomposed the **Surface Temperature Analysis (ST)** epic onto three smaller stories. Label with initial "E" stands for epic.

These three **ST** stories are present in Figure 8, since they also belong to the **ACS** theme. However, there are stories in the **ACS** theme that are not specific only to the **ST** epic (e.g. "*As a climatologist I want to see the results as a plot of individual time series*"), since they are generic stories that can be applied to other epics.

As a first prototype we have worked on a **simplified** version to analyse the surface temperature for one scenario, one model and one ensemble-member:

- Step 1: For a given a set of parameters (one GES scenario, one climate model simulation and one ensemble member), extract the surface temperature field
- Step 4: Plot the surface temperature extracted in a graph

The stories regarding with this simplified use-case were initially labeled as **"To Do"** (see Figure 12). Currently, these three stories have been implemented (see Figure 13), which helped us to test DARE's ability to deliver capacity and functionalities to research developers supporting climatologists. The stories were discussed to identify the aspects of some of the generic features of DARE introduced in Section 5, and details of their implementation can be found at deliverable D7.3 [8].

These impact the generation of provenance information and the transfer of the data towards generic data stores. For instance, e-infrastructure storage services such as B2DROP[5].

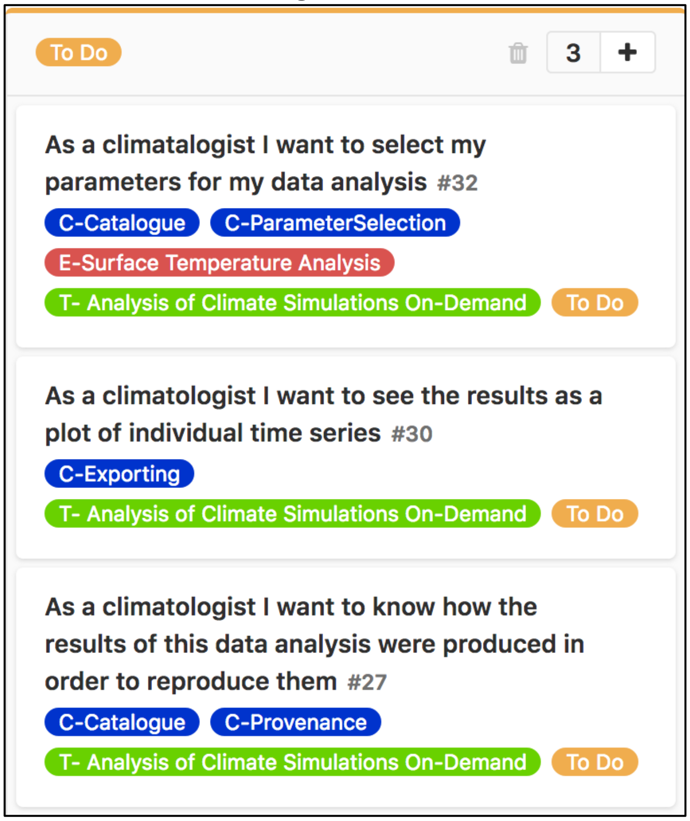


Figure 12. "To do" stories for developed at M9 of the project as the first climate use-case prototype.

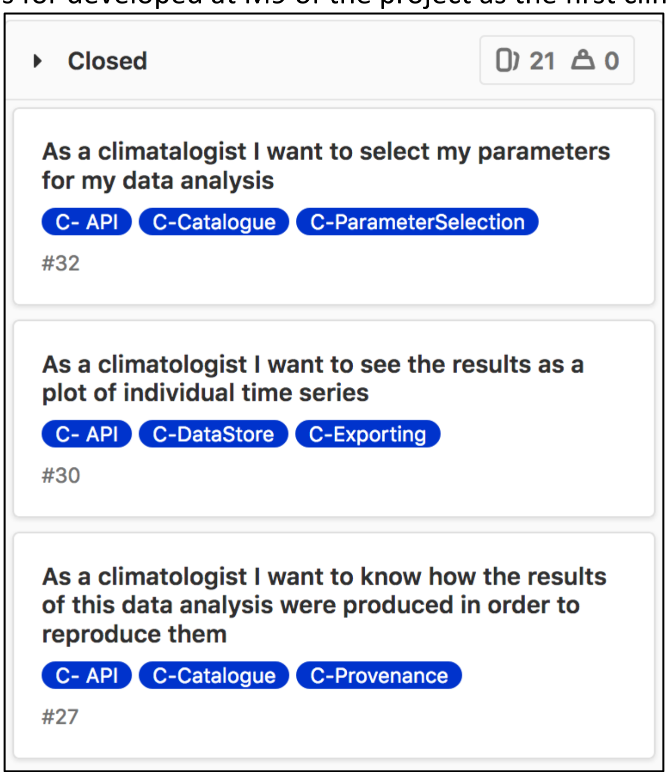


Figure 13. Current stories status (at M18 of the project) of the first climate use-case prototype.

[5] https://www.eudat.eu/services/b2drop

5 DARE Features

Through the analysis of the seismological and climate use cases and the architectural indications, we produced a collection of *General Features*, which should be addressed when implementing the DARE core API. These are listed below:

1) Prepare workflows for provenance and lineage recording
2) Register a workflow to a dedicated repository
3) Prepare input data for a specific workflow run
4) Submit the workflow
5) Move data from the computational cluster to a target resource
6) Monitor the running workflow (domain and system information)

These features went through an initial implementation and will be refined in the following iterations, towards the release of the updated DARE test-bed.

"Feature 1" has been demonstrated already in the first implementation of the use cases. As an example, Figure 14 shows how the customisation of lineage is achieved for a simple function, which is part of a larger workflow in the Rapid Assessment Use case (WP6). Figure 12 instead shows how the provenance requirements are specified by configuration document, which is applied upon submission to the workflow. It shows also the attribution of the function to semantic clusters, inline with the indications of the DARE Architecture, which indicates the DARE Knowledge Base as the source of the definitions of these conceptual classes.

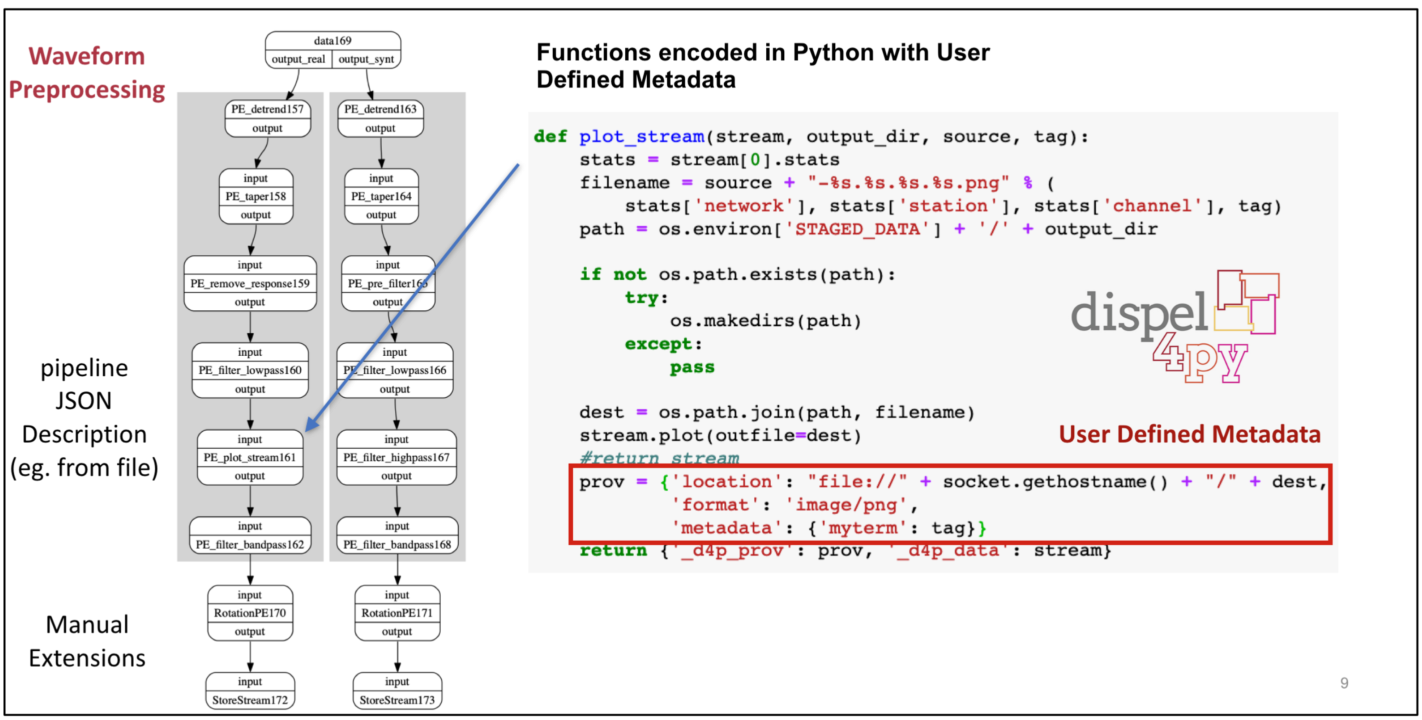


Figure 14. *Waveform pre-processing workflow:* Plotting function that ingests a seismic time-series stream, an output location and a tag describing its synthetic or observed origin. It produces an image and passes the stream to the next processing element The red box highlights how customised metadata are prepared. These will be added to the lineage upon return. Here the volatile time-series is associated with its plot, whose format specified together current location for immediate access and with its origin (described by a user defined metadata term 'myterm').

"Features 2-5" have been also prototyped in this first half of the project. Their implementation includes a service API and a client software interface. We plan to further refine the features aiming at improved useability, thanks to the feedback collected by the implementation of the use cases (thereby the stories already introduced in the previous sections), which have been demonstrated in their training events. Finally, "feature 6" will be further refined and implemented by combining different information and services. These include the runtime-data lineage provided by S-ProvFlow[6] (D3.3 [11])) and the status of the computational nodes that are made available to the users from the underlying Kubernetes-based infrastructure. The lineage services already enable the acquisition and query of the provenance captured at runtime; whose configuration is defined thanks to the "feature 1".

Configuration Profile in JSON with Provenance Types

```
{
     'provone:User': "aspinuso",
     's-prov:description' : "provdemo demokritos",
     's-prov:workflowName': "demo_epos",
     's-prov:workflowType': "seis:preprocess",
     's-prov:workflowId'  : "workflow process",
     's-prov:save-mode'   : 'service'          ,
 # defines the Provenance Types and Provenance Clusters for the Workflow Components
     's-prov:componentsType' :
                           {'PE_taper': {'s-prov:type':(SeismoPE,),
                                         's-prov:prov-cluster':'seis:Processor'},
                            'PE_plot_stream':    {'s-prov:prov-cluster':'seis:Visualisation',
                                               's-prov:type':(SeismoPE,)},
                            'StoreStream':    {'s-prov:prov-cluster':'seis:DataHandler',
                                               's-prov:type':(SeismoPE,)}
                           }}
```

Figure 15. Extract of a provenance configuration json for *Waveform pre-processing* workflow. It shows the assignment of Provenance Type *SeismoPE* and the semantic clustering (property 's-prov:prov-cluster') to relevant processing elements used by the workflow. Abstract components implemented by the same processing element will automatically assume the same provenance setup. *e.g. StoreStream172* and *StoreStream173 of Figure 11*.

6 Conclusions

Use cases from both user communities were analysed in discussions with WP6 and WP7 and broken down into high-level requirements (epics) and more detailed user stories. Issue boards document the themes, epics and stories. During the initial refinement phases the teams gained confidence with the Agile terminology and approaches. This led to the presentation of the User Stories as a Story Board following the same layout and structure for both communities. Weekly Refinement Calls were

[6] https://gitlab.com/project-dare/s-ProvFlow

scheduled to discuss and validate the technical developments by pursuing the implementation of the user stories from WP6 and WP7. Major tasks, planning and directions were discussed during face to face Hackathons held in Athens.

This interaction and commitment to co-design with two demanding user communities provided the first opportunity to test the DARE-architecture proposals set out in D2.1-M12 [5]. With both co-design communities the value of the Workflows-as-a-Service (WaaS) and protected, pervasive persistent provenance (P4) functionalities and subsystems was immediately recognised, because these are already available and used by the communities, and because they are already well integrated. That commitment and engagement is manifest in their plans to write the new functionalities they need in dispel4py, with WP4 delivering needed extensions, and in their plans to use provenance-driven tools from WP3 to organise and accelerate their working practices.

A substantial common core in this architecture has been identified as supporting the two existing user communities as they are represented in DARE. As DARE is a collection of enthusiastic innovators, we recognise that they may not be typical of professionals in such fields. Two activities will help the DARE project address this:

1. Interaction with relevant wider communities at AGU and EGU, and other application-domain events, will collect and collate the reactions representative of typical professional workers; this will be led by WP8.
2. Development of mechanisms for incremental adoption, so that established working practices do not need revising and so that professionals can avoid risks until incentives for adoption are evident; this will be led by WP2.

The DARE architecture once tested and refined by these processes, and the substantial common framework that is tailored for each application, will have the potential for use supporting the work of many data-powered communities. WP2 will identify and explain these benefits in order to prepare the ground for wider adoption; a prerequisite of long-term sustainability.

7 References

[1] Cohn M, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.

[2] *Manifesto for agile software development*, 2001, http://agilemanifesto.org

[3] Sutherland J, Schwaber K, *The Scrum Guide,* 2018, http://scrumguides.org

[4] Brenner P, *A Technical Tutorial on the IEEE 802.11 Protocol, Breezecom Wireless Communications edition,* 1997.

[5] Malcolm Atkinson, et al. *DARE Deliverable D2.1-M12 Dare Architecture and Technical Positioning,* 2018, http://project-dare.eu/wp-content/uploads/2019/03/D2.1-DARE-Architecture-and-Technical-Positioning-I_final_draft.pdf

[6] Christian Pagé, et al. *DARE Deliverable D7.1 Requirements and Test Cases I*, 2018, http://project-dare.eu/wp-content/uploads/2019/03/D7.1-Requirements-and-Test-Cases-I_Final_draft.doc.pdf

[7] Andreas Rietbrock, et al. *DARE Deliverable D6.1 Requirements and Test Cases I*, 2018, http://project-dare.eu/wp-content/uploads/2019/03/D6.1-Requirements-and-Test-Cases-I_final_draft.pdf

[8] Christian Pagé, et al. *DARE Deliverable D7.3 Pilot Tools and Services, Execution and Evaluation Report*

[9] Federica Magnoni, et al. DARE *Deliverable D6.3 Pilot Tools and Services, Execution and Evaluation Report, https://drive.google.com/open?id=11vx0_PlbAH8pU9LIEGSBNXyAIXwkSNxy*

[10] Malcolm Atkinson, et al. *Comprehensible Control for Researchers and Developers facing Data Challenges has been accepted for e-Science 2019*. IEEE eScience 2019, September, San Diego, US [accepted]
[11] Alessandro Spinuso. DARE *Deliverable D3.3 Data Lineage Services (Software Report)*